NASA Contractor Report 172444

(NASA-CR-172444) ANALYSIS OF ESTIMATION ALGORITHMS FOR CDTI AND CAS APPLICATIONS Final Report (Analytical Mechanics Associates, Inc.) 216 p HC A10/MF A01 CSCL 01D G3/06

N85-22387

Unclas 14740

Analysis Of Estimation Algorithms For CDTI And CAS Applications

Tsuyoshi Goka

Analytical Mechanics Associates, Inc.
Mountain View, California 94043

Contract NAS1-16135
April 1985

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

NASA Contractor Report 172444

Analysis Of Estimation Algorithms For CDTI And CAS Applications

Tsuyoshi Goka

Analytical Mechanics Associates, Inc.
Mountain View, California 94043

Prepared for
Langley Research Center
National Aeronautics and Space Administration
under Contract NAS1-16135

National Aeronautics and
Space Administration

Langley Research Center
Hampton Virginia 23665

April 1985

FOREWORD

This effort for analyzing estimation algorithms for CDTI and CAS applications was supported under NASA Contract No. 1-16135 by Langley Research Center, Hampton, VA. The project technical monitor was Roland L. Bowles of Langley Research Center. Technical discussions with Dr. Bowles, James R. Kelly and David H. Williams of Langley Research Center, and Dr. John A. Sorensen of AMA, are gratefully acknowledged.

The opinions expressed in this report are the author's and do not necessarily represent view-points of the sponsoring agency.

ANALYSIS OF ESTIMATION ALGORITHMS FOR CDTI AND CAS APPLICATIONS

Tsuyoshi Goka

Analytical Mechanics Associates, Inc.
Mountain View, California 94043

SUMMARY

The objectives of this project were to analyze and/or to develop estimation algorithms for Cockpit Display of Traffic Information (CDTI) and Collision Avoidance System (CAS) applications. The algorithms are based on actual or projected operational and performance characteristics of an Enhanced TCAS II traffic sensor developed by Bendix and the Federal Aviation Administration.

Three algorithms are examined and discussed. These are horizontal x and y, range and altitude estimation althorithms. Raw estimation errors are quantified using Monte Carlo simulations developed for each application; the raw errors are then used to infer impacts on the CDTI and CAS applications. Applications of smoothing algorithms to CDTI problems are also discussed briefly.

Conclusions are summarized based on the analysis of simulation results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES - Continued

LIST OF FIGURES - Continued

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AGL	Above Ground Level
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
BCAS	Beacon Collision Avoidance System
CAS	Collision Avoidance System
CDTI	Cockpit Display of Traffic Information
CPA	Closest Point of Approach
EFR	Electronic Flight Rules
EHSI	Electronic Horizontal Situation Indicator
FAA	Federal Aviation Administration
INS	Inertial Navigation System
LOT	Level Occupancy Time
LST	Level Switching Time
MFD	Multi-Function Display
MSL	Mean Sea Level
Mode A	An ATCRBS reply format with aircraft identity
Mode C	An ATCRBS reply format containing an altitude report
Mode S	New Selectable data formats used by a Mode S transponder
MOPS	Minimum Operational Performance Standards
NASA	National Aeronautics and Space Administration
NED	north-east-down (a standard coordinate system)
PWI	Proximity (or Pilot) Warning Indicator
RF	Radio Frequency
TCAS	Traffic (or Threat) Alert and Collision Avoidance System
VFR	Visual Flight Rules

I

INTRODUCTION

NASA Langley Research Center is pursuing a research effort concerning the Cockpit Display of Traffic Information (CDTI) concept [1]. The CDTI is a device which presents information to the pilot and crew depicting the status of surrounding traffic including position and velocity states. The traffic information is provided by a "traffic sensor." Most promising candidate sensors are FAA developed Traffic Alert and Collision Avoidance Systems (TCAS).

TCAS is strictly an airborne system which provides the aircraft separation protection information independent of the ground ATC system. The FAA plans call for developing two types of TCAS--TCAS I and TCAS II. Within each category, a certain latitude in capability is allowed to satisfy a wide spectrum of user requirements. The enhanced TCAS II which is capable of obtaining relative bearing measurements may be able to support CDTI applications. There are two designs in this enhanced TCAS II category. One design developed by MIT/Dalmo Victor is based on the so-called active Beacon Collision Avoidance System (BCAS). The other developed by Bendix is based on the so-called full BCAS concept. The former unit is being tested in actual commercial flight operation environments; and the other is undergoing an extensive flight test with the prototype system.

The current effort is a part of parallel efforts consisting of:

(a) Development of a realistic enhanced TCAS II simulation model, and;

(b) Analysis of the TCAS estimation algorithms for the CDTI and CAS applications.

The companion report, "Enhanced TCAS II/CDTI Traffic Sensor Digital Simulation Model and Program Description [2], contains a detailed discussion of the Bendix designed system. A shorter version of TCAS II description is given in Appendix A.

The objectives of the current effort are to analyze and/or develop estimation algorithms for CDTI and possibly for Collision Avoidance System (CAS) logic applications based on actual or projected operational and performance characteristics. For purposes of estimation three coordinate axes are considered separately. These are x and y, range and altitude. Within these axes, variations in signal configurations, signal sources or algorithm implementation items are examined.

The following procedure is used to compare and analyze the estimation algorithms. The raw estimation errors are quantified using Monte Carlo simulation method. The raw error data are then used to infer impacts on the CDTI and CAS applications. For example, altitude and altitude rate estimation errors per se do not mean much; however, if these are factored into the projected altitude error, then the latter would have a great significance in terms of safe altitude separation threshold of CAS logic.

In Chapter II, basic performance of horizontal x-y filter algorithms are discussed. The basic question in this chapter is how and what kinds of other signals (in addition to the relative range and bearing measurements) are best utilized to provide better estimates in horizontal x-y axes. This is motivated by the fact that the dynamic lag due to maneuvers by Own or target aircraft induces large and sustained errors in position and velocity estimates. These errors can be compensated by utilizing maneuver parameters (such as heading angles) in estimation algorithm. This assumes that the target data are made available via the Mode S data link capability. Also, the questions of filter gains determination are addressed. The filter gains depend on many operational factors; thus, it is not a trivial matter. These questions are probed by means of error statistics generated by Monte Carlo simulation program.

In Chapter III, range and range rate estimates are obtained in several ways. Raw error statistics for each are obtained by Monte Carlo method using "realistic" encounter scenarios. These are, in turn, compared and analyzed in terms of accuracy.

In Chapter IV, an altitude tracker algorithm is developed and presented. The altitude axis poses a special estimation challenge in that the target altitude measurements are quantized to the nearest 100 ft. This causes a certain observability problem. The algorithm is based on the level switching time detection concept. The performance analysis are carried out by comparing the estimation errors with those of a non-quantized alpha-beta tracker. The latter represents the best possible without aiding the estimation algorithm with external signals.

In Chapter V, the raw error statistics (obtained in the previous three chapters) are analyzed from the user's view-point, i.e., from the CDTI and CAS logic application aspects. These would provide relative and absolute merits of particular estimation algorithms with respect to the design requirements. Also, a short discussion of smoothing (rather than estimation) algorithms are given with respect to CDTI applications.

Appendices A through D provide peripheral but important information which are directly related to this effort.

II

HORIZONTAL X-Y FILTERS

Filter Configuration

In the context of an airborne CDTI sensor based on the proposed enhanced TCAS II, the relative position of an intruder aircraft is obtained in range and bearing axes. The verticle axis is provided by an encoding altimeter (either above MSL or AGL). The range and bearing measurements are taken with respect to a cylindrical coordinate system attached to Own fuselage; therefore, the coordinate system is itself subject to translational and rotational motions as Own aircraft undergoes maneuvers.

The sensor is designed to account for Own's orientation effect by means of software compensation utilizing onboard INS generated attitude angles. (A brief description of the sensor surveillance and operational characteristics is given in Appendix A.) For filter analysis purposes, point-mass kinematics are assumed for both Own and Target aircraft. That is, that the effect of Own's orientation angles is assumed to be negligible. This may be justified by the fact that (a) an INS provides accurate orientation, and (b) low frequency bias errors do not affect velocity estimates. If the altitudes are ignored, the relative horizontal measurements (range and bearing) are given by

$$r_m = [\Delta x^2 + \Delta y^2]^{\frac{1}{2}} + \xi_r \ , \qquad (1)$$

$$b_m = \tan^{-1}(\Delta y/\Delta x) + \xi_b \ ,$$

where ξ_r and ξ_b are measurement errors.

For the purpose of designing estimation algorithms, a model is needed to describe the relative kinematics. Now, the magnitude of purely longitudinal acceleration, (i.e., along the velocity vector) is small - usually less than 2 - 3 kt/sec for commercial operation. Thus, the longitudinal acceleration effect (approximately 2.5 ft in position change at 3 kt/sec) is masked by somewhat large measurement errors (ranging error of 100 feet).

However, the acceleration due to a turn is not so trivial.

A 1/2 g acceleration corresponding to the bank angle of 25 deg is not uncommon for commercial operation. A suitable kinematic model is obtained by assuming that aircraft follow a series of straight line or circular arc segments. If position and velocity vectors, p and v, are defined with respect to an earth fixed rectangular coordinate system, then each aircraft is described by an equation of the form

$$\frac{d}{dt}\begin{bmatrix} P_i \\ v_i \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & \Omega_i \end{bmatrix} \begin{bmatrix} P_i \\ v_i \end{bmatrix} \quad , \quad i = (0) \text{ Own or Target (T)} \qquad (2)$$

where

$$P_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad , \quad v_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \quad , \quad \Omega_i = \begin{bmatrix} 0 & \omega_i \\ -\omega_i & 0 \end{bmatrix} \quad , \text{ and } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

The turn rate, ω_i is a piece-wise constant time function. By subtracting, the relative kinematic equation for two aircraft can be expressed as

$$\frac{d}{dt}\begin{bmatrix} \Delta p \\ \Delta v \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta v \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta a \end{bmatrix} \qquad (3)$$

where Δa is the relative acceleration given by

$$\Delta a = \Omega_T v_T - \Omega_0 v_0 = \Omega_T \Delta v + (\Omega_T - \Omega_0) v_o \ . \qquad (4)$$

Obviously when both aircraft are non-accelerating (straight line flight), then $\Delta a = 0$, or Δv = constant.

Equations (1) and (3) form the basis for designing horizontal x-y filters to estimate position and velocity. In the following sections several filter algorithms are derived and discussed. The configuration differences are based on different signals available to aid the relative position measurement. These include:

(1) no additional measurements are available;

(2) Own body rates are available; and

(3) Own and Target body rates are available.

In cases (2) and (3) it is assumed that Own or Target acceleration can be derived to compensate for the unknown acceleration components. In case (3), availability of suitable avionics and an air-to-air communication link is assumed. Because a Mode S transponder is an integral element of a TCAS, it is only necessary to establish a digital data link between an onboard sensor such as an INS or a navigation computer to the so-called Standard Message Interface port of the Mode S transponder. It is noted that the Own signal compensation is relatively easy in the sense that a digital data interface between an INS and TCAS is in place. Cross-feed signals include the three Euler angles ϕ, θ and ψ and the ground speed, V_G. Own acceleration can be generated easily from these available signals.

Non-Aided Filter Configuration In the absence of any acceleration indication, there are essentially two approaches for designing estimation algorithms. One is simply to ignore the acceleration input. The other is to estimate the unknown acceleration. The second approach is generally very difficult. Reference [3,4] discusses several methods applied to a simpler problem of tracking maneuvering aircraft using a ground based Mode S sensor. Compared to the previous ground based study, there are major differences in TCAS surveillance functions :

(1) ground-based versus aircraft based ;

(2) sampling period of 4-5 sec vs 1-8 sec; and,

(3) bearing error of 0.04 deg vs 1-2 deg.

The most crucial airborne disadvantage is the bearing error which is 25-50 times larger in magnitude. The linear equivalance is 65 ft compared to 1600-3200 ft at a range of 15 nmi. Therefore, attempting to estimate unknown acceleration in the given noise environment is not realistic. Thus, the first approach is now studied further.

If the acceleration is assumed to be zero, then Eq (3) reduced to

$$\frac{d}{dt}\begin{bmatrix}\Delta p\\ \Delta v\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta p\\ \Delta v\end{bmatrix} . \tag{5}$$

By discretizing over one sample time interval, Δt, and by writing the x component equation (y equation is identical), then it follows

$$\begin{bmatrix}\Delta x\\ \Delta \dot{x}\end{bmatrix}_{n+1} = \begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta \dot{x}\end{bmatrix}_{n} . \tag{6}$$

The pesudo measurement equations are

$$\begin{aligned}\Delta x_n^m &= r_n^m \cos b_n^m \simeq \Delta x_n + \xi_x ,\\ \Delta y_n^m &= r_n^m \sin b_n^m \simeq \Delta y_n + \xi_y .\end{aligned} \tag{7}$$

Using Eqs. (6) and (7), the standard filter algorithm is given by

$$\begin{aligned}&\text{position prediction:} && \Delta x^+ = \hat{\Delta x}_n + \Delta t \cdot \hat{\Delta \dot{x}}_n ;\\ &\text{measurement error:} && \tilde{\Delta x} = \Delta x_{n+1}^m - \Delta x^+ ; \text{ and}\\ &\text{estimate update:} && \hat{\Delta x} = \Delta x^+ + g_{x1}\cdot\tilde{\Delta x} ,\\ & && \hat{\Delta \dot{x}} = \hat{\Delta \dot{x}} + g_{x2}\cdot\tilde{\Delta x}\end{aligned} \tag{8}$$

where g_{x1} and g_{x2} are filter gains. Note that Eq. (8) would represent an optimal filter if Own and Target aircraft are not maneuvering. Choosing proper filter gains is more of an art than a science. There are many factors involved such as noise variance reduction, dynamic error minimization, computational ease, and so on. This is discussed in a later section.

Own Signal-Aided Filter Configuration When the Own acceleration signal is available, at least half of the acceleration term in the dynamic equation can be compensated. Thus, after proper discretization, the model equation equivalent to (6) is given by

$$\begin{bmatrix} \Delta x \\ \dot{\Delta x} \end{bmatrix}_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \dot{\Delta x} \end{bmatrix}_{n} - \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} \hat{\ddot{x}}_{0,n} \tag{9}$$

where $\hat{\ddot{x}}_{0,n}$ is the estimated acceleration from Own onboard signals. Taking advantage of an inertial grade INS, differential ground velocity, $\hat{\delta v}_x$ can be used.

$$\hat{\delta v}_{x,n} = \hat{\dot{x}}_{0,n} - \hat{\dot{x}}_{0,n-1} \simeq \Delta t \cdot \hat{\ddot{x}}_{0,n} \ . \tag{10}$$

The estimation algorithm is given by

$$\text{state prediction:} \quad \Delta x^{+} = \hat{\Delta x}_n + \Delta t \cdot \hat{\dot{\Delta x}}_n - (\Delta t/2)\, \hat{\delta v}_{x,n} \ ,$$

$$\dot{\Delta x}^{+} = \hat{\dot{\Delta x}}_n - \hat{\delta v}_{x,n} \ ;$$

$$\text{measurement error:} \quad \tilde{\Delta x} = \Delta x^{m}_{n+1} - \Delta x^{+} \ ; \text{ and} \tag{11}$$

$$\text{estimate update:} \quad \hat{\Delta x}_{n+1} = \Delta x^{m}_{n+1} + g_{x1} \cdot \tilde{\Delta x} \ ,$$

$$\hat{\dot{\Delta x}}_{n+1} = \dot{\Delta x}^{+} + g_{x2} \cdot \tilde{\Delta x} \ ,$$

where g_{x1} and g_{x2} are again the filter gains. As expected, this filter configuration would be ideally suited if the Target aircraft does not accelerate. Also, an intuitive expectation that this filter would be "twice as good" as the previous configuration because it compensates for at least one half of the problem causes is not correct. Even though this is true most of the time, it is apparent from Eqs (3) and (4) that if the relative acceleration $\Delta a \simeq 0$ even though a_0 and a_T are large, then the Non-Aided Configuration would outperform the Own-Aided Configuration.

Own and Target Signal-Aided Filter Configuration When Own and Target signals are both available, then most of the unknowns in the system disappear. For the purpose of algorithm design, it is assumed that the Target cross-links its ground speed V_{GT} and heading ψ as part of its Mode S surveillance reply message. (If the Target is equipped with an enhanced TCAS II, then these signals are available per specification, i.e., no hardware modification is necessary.) Under these assumptions, Target differential ground velocity can be computed as

$$\hat{\delta v}_{Tx,n} = v_{GT,n} \cos(\psi_{T,n}) - v_{GT,n-1} \cos(\psi_{T,n-1}) \simeq \Delta t \cdot \hat{\ddot{x}}_{Tn} ,$$

(12)

$$\hat{\delta v}_{Ty,n} = v_{GT,n} \sin(\psi_{T,n}) - v_{GT,n-1} \sin(\psi_{T,n-1}) \simeq \Delta t \cdot \hat{\ddot{y}}_{Tn} .$$

It is noted that the sampling time for Own signals and that of Target signals may not be the same, i.e., they are not synchronous. Target data are available only when the interrogation/reply cycle is complete, which may take from 1 to 8 sec; Own signals are available at the TCAS basic cycle time of 1 sec. For example, Target data may be available every 4 sec compared to Own data available every 1 sec. This implies that special care is needed in processing Own inertial data.

If the computed acceleration terms are incorporated, the discrete system equation becomes

$$\begin{bmatrix} \Delta x \\ \Delta \dot{x} \end{bmatrix}_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \dot{x} \end{bmatrix}_n + \begin{bmatrix} \Delta t/2 \\ 1 \end{bmatrix} \delta v_{Tx,n} - \begin{bmatrix} \Delta t/2 \\ 1 \end{bmatrix} \hat{\delta v}_{Ox,n} .$$

(13)

The corresponding estimation algorithm is given by

state prediction:
$$\Delta x^+ = \Delta \hat{x}_n + \Delta t \cdot \Delta \hat{\dot{x}}_n + (\Delta t/2) [\hat{\delta v}_{Tx,n} - \hat{\delta v}_{Ox,n}] ,$$

$$\Delta \dot{x}^+ = \Delta \hat{\dot{x}}_n + \hat{\delta v}_{Tx,n} - \hat{\delta v}_{Ox,n} ,$$

measurement error:
$$\tilde{\Delta x} = \Delta x^m_{n+1} - \Delta x^+ \quad ; \text{ and} \qquad (14)$$

estimate update:
$$\Delta \hat{x}_{n+1} = \Delta x^+_n + g_{x1} \tilde{\Delta x} ,$$

$$\Delta \hat{\dot{x}}_{n+1} = \Delta \dot{x}^+_n + g_{x2} \tilde{\Delta x} ,$$

where g_{x1} and g_{x2} are filter gains.

As far as the estimation algorithms are concerned, there are very small computational differences. The essential difference is in computing the predicted state; the rest is identical. The real computational loads for the aided configurations lie in pre-processing and keeping track (time- and book-keeping) of Own and Target inertial signals. These seem rather trivial compared to other more complicated processing performed in other TCAS modules.

Filter Gain Generation

Once the configuration is chosen, the filter gains need to be specified in order to implement the algorithm. In this section, three methods of selecting the gain values are discussed. These are (a) fixed αβ tracker gain; (b) Kalman filter gain, and (c) table-look-up gain. These methods are analysed for each of the three configurations.

Method (a) would be simplest from the computational point of view. The fixed gain configuration would have a major disadvantage: the fixed filter gains imply the noise reduction ratio remains the same. Thus, the "roughness" of the estimate would be proportional to the "roughness" (or noise magnitude) of the input signal. As is well-known, the noise variance of the pseudo x or y measurement is affected by the so-called range effect due to the basic radar coordinate system. Thus, the input variance would be proportional to range; hence, the estimate error variances would be proportional to range.

Method (b) would be most suitable in terms of best performance in the sense that gains are automatically adjusted according to input noise variance. This feature would "desensitize" the range effect problem encountered by the first method. The price for the added performance gain is additional computational load.

Method (c) tries to strike a mid-point in performance and in computational load. Very briefly, the filter time constant (approximately the reciprocal of the filter bandwidth) parameter is stored in a two-parameter look-up-table of the sampling period and measurement noise standard devia-

tion level. The time constant is used to compute position and velocity feedback gains. For convenience, important equations are repeated in Table 1.

A few points need to be noted. The estimated acceleration, $\hat{a}_n$ in Eq (15) is 0 for non-aided configuration; Own estimated acceleration, $\hat{a}_{On}$ for Own-aided configuration; or $\hat{a}_{Tn} - \hat{a}_{On}$ for Own-and-Target-aided configuration. The model dynamics and state prediction Eqs (15) and (17) are identical; therefore, the modeling error is dominated by the acceleration error,

$$\tilde{a}_n = \hat{a}_n - a_n \quad . \tag{21}$$

As pointed out previously, the pseudo measurement errors ξ_{xn} (and ξ_{yn}) in Eq (16) would have the covariance matrix

$$E\left(\begin{bmatrix}\xi_x\\ \xi_y\end{bmatrix}\begin{bmatrix}\xi_x & \xi_y\end{bmatrix}\right) = \begin{bmatrix}\sigma_x^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_y^2\end{bmatrix} \tag{22}$$

$$= \begin{bmatrix}\cos^2 b\,\sigma_r^2 + r^2\sin^2 b\,\sigma_b^2 & \cos b \sin b\,\sigma_r^2 - r^2\sigma_b^2\\ \cos b \sin b\,(\sigma_r^2 - r^2\sigma_b^2) & \sin^2 b\,\sigma_r^2 + r^2\cos^2 b\,\sigma_b^2\end{bmatrix} .$$

Here σ_r and σ_b are range and bearing error standard deviations. Note that the off-diagonal term is generally non-zero.

Method (a) - Fixed Gains This is the method used to obtain the α and β tracker gains in the current TCAS design. Therefore, the method is directed toward the non-aided configuration; however, the design procedure can be applied to other configurations. The following discussion follows Ref. [5] very closely. The basic idea is to compute the error due to measurement error and the error due to acceleration error. If these errors are combined statistically in a correct way, then we can optimize the gain values to minimize the total error.

(i) Error due to measurement error only

Table 1. Summary of Estimation Equations

Model	Dynamic Eqn.	$\begin{bmatrix}\Delta x \\ \Delta\dot{x}\end{bmatrix}_{n+1} = \begin{bmatrix}1 & \Delta t \\ 0 & 1\end{bmatrix}\begin{bmatrix}\Delta x \\ \Delta\dot{x}\end{bmatrix}_n + \begin{bmatrix}\Delta t^2/2 \\ \Delta t\end{bmatrix} a_{x,n}$	(15)
	Measurement Eqn.	$\Delta x_n^m = r^m \cos b_n^m \simeq \Delta x_n + \xi_{x,n}$ $\Delta y_n^m = r^m \sin b_n^m \simeq \Delta y_n + \xi_{y,n}$	(16)
Estimation	State Prediction	$\begin{bmatrix}\Delta x \\ \Delta\dot{x}\end{bmatrix}^+ = \begin{bmatrix}1 & \Delta t \\ 0 & 1\end{bmatrix}\begin{bmatrix}\Delta\hat{x} \\ \Delta\dot{\hat{x}}\end{bmatrix}_n + \begin{bmatrix}\Delta t^2/2 \\ \Delta t\end{bmatrix}\hat{a}_{x,n}^*$	(17)
	Measurement Error	$\Delta\tilde{x} = \Delta x_n^m - \Delta x^+$	(18)
	State Update	$\begin{bmatrix}\Delta\hat{x} \\ \Delta\dot{\hat{x}}\end{bmatrix}_{n+1} = \begin{bmatrix}\Delta x \\ \Delta\dot{x}\end{bmatrix}^+ + \begin{bmatrix}\alpha \\ \beta/\Delta t\end{bmatrix}\Delta\tilde{x}$	(19)
Estimation Error		$\begin{bmatrix}\Delta\tilde{x} \\ \Delta\dot{\tilde{x}}\end{bmatrix}_{n+1} = \begin{bmatrix}1-\alpha & \Delta t(1-\alpha) \\ -\beta/\Delta t & 1-\beta\end{bmatrix}\begin{bmatrix}\Delta\tilde{x} \\ \Delta\dot{\tilde{x}}\end{bmatrix}_n + \begin{bmatrix}(1-\alpha)\Delta t^2/2 \\ (2-\beta)\Delta t/2\end{bmatrix}\tilde{a}_n^{**} + \begin{bmatrix}\alpha \\ \beta/\Delta t\end{bmatrix}\xi_{x,n+1}$	(20)

* $\hat{a}_{x,n} = 0$ (Non-aided) ; $= -\hat{a}_{0x,n}$ (Own-aided); and
$= \hat{a}_{Tx,n} - \hat{a}_{0x,n}$ (Own and Target-aided)

** $\Delta\tilde{x} \equiv \Delta\hat{x} - \Delta x$; $\Delta\dot{\tilde{x}} \equiv \Delta\dot{\hat{x}} - \Delta x$ (state estimation error)
$\tilde{a}_n \equiv \hat{a}_{xn} - a_{xn}$ (acceleration model error)

If the acceleration error, $\tilde{a}_n$ in Eq (21) is set to zero, and we solve for the steady-state error covariance using the standard linear covariance propagation formula (assuming σ_x^2 is stationary), then the following relationships are obtained:

$$\sigma_{\tilde{x}}^2 = \left[\frac{2\alpha^2 + \beta(2-3\alpha)}{\alpha(4-\beta-2\alpha)}\right]\sigma_x^2 \quad ,$$

$$\sigma_{\dot{\tilde{x}}}^2 = \frac{1}{\Delta t^2}\left[\frac{2\beta^2}{\alpha(4-\beta-2\alpha)}\right]\sigma_x^2 \quad . \tag{23}$$

These two formulas express the estimation errors in terms of input error variance, sampling period and the filter gains.

(ii) Error due to acceleration model error.

Now, set the input noise to zero. Then, Eq (21) expresses the estimation errors due only to the acceleration error. Since the nature of this error is low frequency, we can compute "worst" type error due to "worst" possible acceleration error. By solving for the steady state error, given $\tilde{a}_M$ as the maximum model error, the following relationships are obtained :

$$\tilde{x}_{ss} = (\Delta t)^2 \frac{1-\alpha}{\beta} \tilde{a}_M \quad ,$$

$$\dot{\tilde{x}}_{ss} = \frac{\Delta t}{2} \frac{2\alpha-\beta}{\beta} \tilde{a}_M \quad . \tag{24}$$

These two formulas express the steady state errors in terms of maximum acceleration uncertainty, sampling period and the filter gains.

(iii) Total Errors

A natural way to combine the statistical errors of Eq (23) and the deterministic errors of Eq (24) is to take the root mean square. The individual errors can be interpreted as Eq (24) being the mean error and Eq (23) being the variance about the mean. Thus

$$\tilde{x}_{rms} = \left[\frac{2\alpha^2 + \beta(2-3\alpha)}{\alpha(4-\beta-2\alpha)} \sigma_x^2 + (\Delta t)^4 \frac{(1-\alpha)^2}{\beta^2} \tilde{a}_M^2 \right]^{1/2} ,$$

$$\tilde{\dot{x}}_{rms} = \left[\frac{1}{\Delta_t^2} \frac{2\beta^2}{\alpha(4-\beta-2\alpha)} \sigma_x^2 + \frac{(\Delta t)^2}{\beta^4} \frac{(2\alpha-\beta)^2}{\beta^2} \tilde{a}_M^2 \right]^{1/2} . \qquad (25)$$

Formulas (25) express the total estimation errors in terms of measurement noise magnitude (σ_x), maximum acceleration uncertainty ($\tilde{a}_M$), the sampling period (Δt) and the filter gains (α and β). Therefore, given suitable numbers for σ_x, $\tilde{a}_M$ and Δt, the best α and β can be obtained which minimizes the rms errors. In Ref. [5], this was done by using σ_x = 825 ft, and $\tilde{a}_M = 0.5g = 16.1$ fps^2 for $\Delta t = 1, \ldots, 8$ sec. Table 2 shows the "optimum" gains.

Table 2. Optimum Gains for Non-Aided Configuration

Δt (sec)	1	2	3	4	5	6	7	8
α	0.25	0.37	0.465	0.53	0.58	0.62	0.645	0.665
$\beta/\Delta t$ sec^{-1}	0.066	0.0875	0.1	0.1078	0.113	0.114	0.124	0.114
$\sigma_{\tilde{x}}$ (ft)	406	506	570	615	650	676	713	719
$\sigma_{\tilde{\dot{x}}}$ (fps)	83.1	95.5	102.8	109.0	114.8	117.4	132.4	122.7
$\tilde{x}_{ss}$ (ft)	183	232	258	281	299	322	323	379
$\tilde{\dot{x}}_{ss}$ (fps)	52.9	70.2	50.7	47.0	42.4	39.1	27.6	29.7
$\tilde{x}_{rms}$ (ft)	445	557	626	676	716	749	783	813
$\tilde{\dot{x}}_{rms}$ (fps)	98.5	118.5	114.6	118.7	122.4	123.8	135.2	126.2

The value of σ_x = 825 ft was chosen, because this number has been "traditionally" important in the airborne based collision avoidance concept [6]. The maximum acceleration $\tilde{a}_M = 0.5g$ was selected based on the maximum appearing in the so-called MITRE's FAA basic model [7]. Some comments are needed.

A σ_x of 825 ft implies that the target range is 7.8 nmi if bearing error is 1 deg and 3.9 nmi if 2 deg. Beyond these ranges, the actual linear error would be larger (e.g., at 10 nmi range and 2 deg bearing error, σ_x = 2100 ft). Within these ranges the linear error would be smaller (e.g., at 2 nmi range and 1 deg bearing error, σ_x = 210 ft). Therefore, the range effect needs to be accounted for.

In the steady state error derivation, it was assumed that the maximum acceleration was maintained indefinitely. This assumption represents the worst possible case. From Table 2 it is clear that the position and velocity errors are dominated by the measurement noise magnitude but not by the dynamic error. This seems to indicate that the gain values could be made a little lower at this measurement error level. However, it should be noted that the above gains were derived with Own being stationary. Thus, if Own aircraft is also conducting a turning maneuver, the acceleration uncertainty could be larger.

Similar analysis may be pursued for other configurations. But it would be more effective to proceed to the other two gain selection methods.

Method (b) - Kalman Filter Gains One of the major disadvantages of a fixed gain filter in a radar environment is that the range effect is not automatically accounted for. That is, "optimized" gains at the error level of σ_x = 825 ft (range of 3.9 nmi at σ_b = 2 deg) is optimal at that point and suboptimum everywhere else. This is the main motivation for utilizing gains based on the Kalman filter theory.

There are two difficulties with this approach which need to be discussed. One is the treatment of acceleration uncertainty. The other is the coupling problem of x and y axes due to non-zero covariance ($\sigma_{xy} \neq 0$) in the pseudo-measurement errors. The latter problem can be significant to the extent that the state covariance should be a 4 x 4 rather than two 2 x 2 matrix.

For the sake of saving computations, this statistical dependence is ignored, and the filter gains are solved only for decoupled axis.

The problem of unknown acceleration is not trivial. In the usual application, the process noise magnitude is varied to "tune" the Kalman filter. Therefore, the unknown acceleration is assumed to be a white noise process, and the magnitude is varied so that the gains yield a satisfactory filter performance. For the purpose of obtaining the Kalman filter equation, the acceleration input in Eq (15) is assumed to be a zero-mean white noise process with standard deviation of σ_a. (Equation (17) needs to be modified accordingly.) Furthermore, the measurement error is approximated by

$$\sigma_x^2 = r^2 \sin^2 b \, \sigma_b^2 \quad ; \quad \sigma_y^2 = r^2 \cos^2 b \, \sigma_b^2 \quad . \tag{26}$$

With these simplifying assumptions, the usual covariance equations for linear systems can be developed. For the two state filter, the equations are particularly simple [8].

Covariance Prediction (Propagation):

$$\begin{aligned} p_1^+ &= p_1 + 2\Delta t \, p_2 + \Delta t^2 \, p_3 + (\Delta t^3/3)\sigma_a^2 \quad , \\ p_2^+ &= p_2 + \Delta t \, p_3 + (\Delta t^2/2)\sigma_a^2 \quad , \\ p_3^+ &= p_3 + \Delta t \, \sigma_a^2 \quad . \end{aligned} \tag{27}$$

Gain Computation:

$$\begin{aligned} \alpha_n &= p_1^+ \, (p_1^+ + \sigma_x^2)^{-1} \\ \beta_n/\Delta t &= p_2^2 \, (p_1^+ + \alpha_x^2)^{-1} \end{aligned} \tag{28}$$

Covariance Update:

$$p_1 = \alpha_n \sigma_x^2$$

$$p_2 = (\beta_n/\Delta t)\sigma_x^2 \qquad (29)$$

$$p_3 = p_3^+ - (\beta_n/\Delta t)^2 (p_1^+ + \sigma_x^2)^{-1}$$

The terms are defined as

$$p_1 = E(\tilde{\Delta x}^2) \quad , \quad p_2 = E(\tilde{\Delta x}\ \dot{\tilde{\Delta x}}) \quad , \quad \text{and} \quad p_3 = E(\dot{\tilde{\Delta x}}^2) \ ,$$

and $(.)^+$ indicates the predicted value according to the usual linear system propagation formula. Time reference is suppressed in the above; however, if the equations are coded in the exact sequence appearing above, then the recursive nature of algorithm is maintained. Furthermore, the p_i^+'s can be stored in the p_i's locations.

The conclusion is clear. These computations are simple enough that implementation in a micro-processor should be straight forward. In an actual implementation, some other considerations need to be made:

(1) Equation (26) cannot be used to compute σ_x^2 and σ_y^2 because the true values of r and b a e not known. Instead they are approximated by

$$\sigma_x^2 \doteq (\Delta y_n^m)^2 \sigma_b^2 \quad ; \quad \sigma_y^2 \doteq (\Delta x_n^m)^2 \sigma_b^2 \quad .$$

(2) Variances are carried instead of standard deviations. Thus, actual numerical values may be quite large. This means that the so-called round-off or over-flow problems arising from a limited word size computer need to be addressed.

(3) Sometimes it is useful to limit the values of σ_x (σ_y) between a minimum and a maximum. This will prevent the gains from becoming too high or too low.

Table 3 shows the Kalman gains and estimation error standard deviations depending on the input noise magnitude, σ_a. The measurement error magnitude was 825 ft. When Table 3 is compared with Table 2, the general trend is clearly the same; i.e., gains become larger as the sampling time becomes longer. The proportional gain (α) seems to be very close (especially the σ_a = 32 fps case); however, the Kalman velocity gain is at least 2 to 5 times smaller. This implies that using the Kalman filter algorithm to determine the filter gains is not unreasonable, if the acceleration noise magnitude was used to tune the performance.

Figure 1 shows the "steady state" Kalman gains as functions of range. Along side the gains, the measurement and estimation error standard deviations are plotted. Figure 1 (a and b) corresponds to the bearing error of 1 deg and 1 and 4 sec sampling periods. Figure 1 (c and d) corresponds to the error magnitude of 2 deg at the same sampling periods. As can be seen, the x error is proportional to range, whereas the corresponding estimation error magnitude becomes large at the far range, the filter gains are lowered proportionally to reduce the "total" error magnitude (balancing the acceleration uncertainty and the position uncertainty).

Method (c) - Table-look-up Gains This method is a compromise between the last two methods. The first method is an optimization based on a numerical minimization procedure. The Kalman algorithm is more of an analytical method. The art resides in choosing the input (acceleration) uncertainty. Basically, the filter gain optimization depends on four parameters: (1) filter time constant, τ_f ; (2) sampling period, Δt ; (3) measurement error magnitude, σ_x ; and (4) acceleration input uncertainty, $\tilde{a}$. That is, the estimation error is given by functions of the form

$$\begin{aligned} \tilde{x}_{rms} &= f_1(\tau_f, \Delta t, \sigma_x, \tilde{a}_x) \quad , \\ \dot{\tilde{x}}_{rms} &= f_2(\tau_f, \Delta t, \sigma_x, \tilde{a}_x) \quad . \end{aligned} \tag{30}$$

Thus, the objective is to minimize $\tilde{x}_{rms}$ (or $\dot{\tilde{x}}_{rms}$) with respect to the filter time constant, τ_f, from which gains may be computed. The difficulty is that $f_1(\cdot)$ and $f_2(\cdot)$ are not analytically tractable for a minimization procedure.

Table 3. Kalman Gains and Error Standard Deviations

σ_x = 825 ft, σ_a = 8 fpss

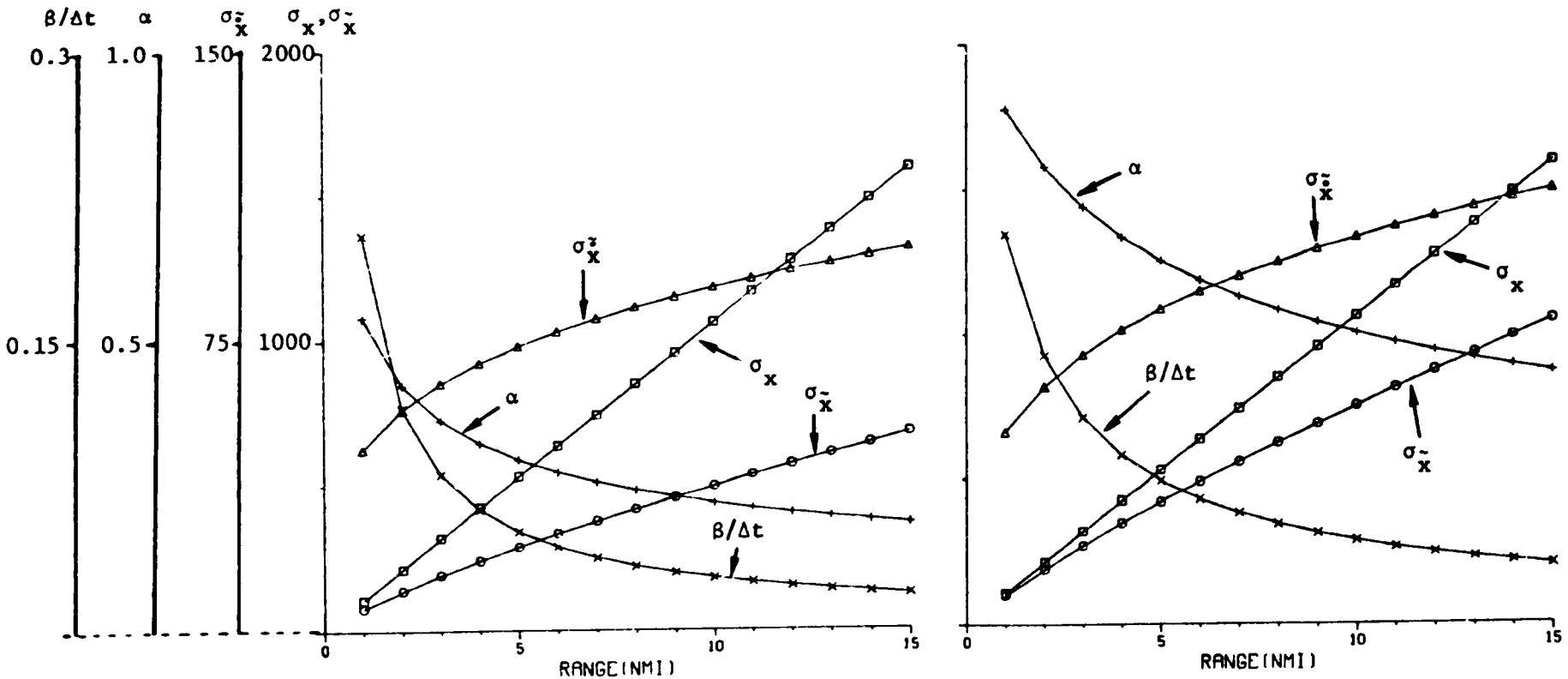
Δt (sec)	1	2	3	4	5	6	7	8
α	0.184	0.247	0.272	0.326	0.372	0.414	0.451	0.484
$\beta/\Delta t$ sec^{-1}	0.015	0.015	0.014	0.016	0.017	0.018	0.019	0.02
σ_x (ft)	353	410	430	471	503	531	554	574
$\sigma_{\dot{x}}$ (fps)	36.6	34.1	33.5	34.4	35.0	35.6	36.0	36.3

σ_x = 825 ft, σ_a = 16 fpss

Δt (sec)	1	2	3	4	5	6	7	8
α	0.179	0.282	0.362	0.427	0.482	0.53	0.572	0.608
$\beta/\Delta t$ sec^{-1}	0.018	0.023	0.027	0.029	0.031	0.033	0.034	0.034
σ_x (ft)	349	438	496	539	573	601	624	643
$\sigma_{\dot{x}}$ (fps)	49.8	53.4	55.4	56.7	57.6	58.3	58.8	59.2

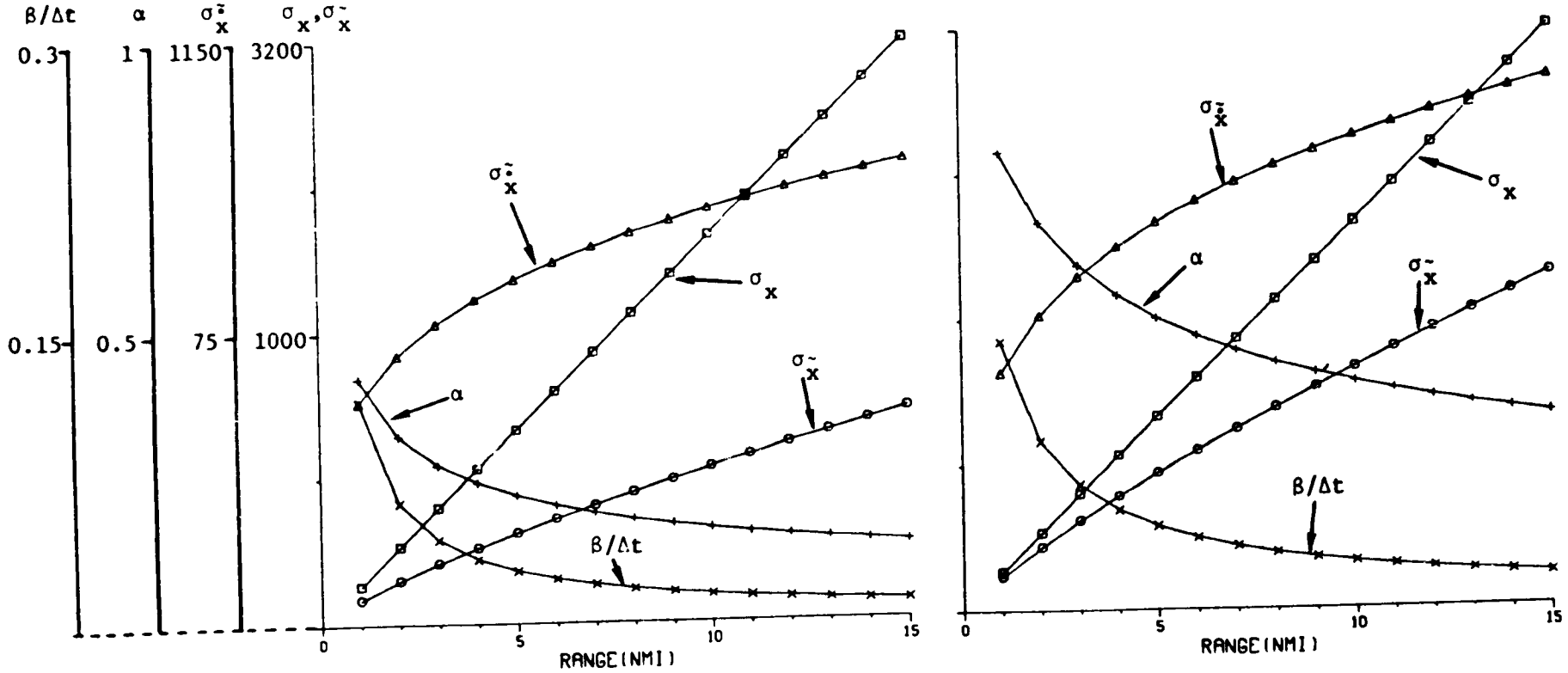
σ_x = 825 ft, σ_a = 32 fpss

Δt (sec)	1	2	3	4	5	6	7	8
α	0.243	0.374	0.47	0.545	0.606	0.656	0.698	0.734
$\beta/\Delta t$ sec^{-1}	0.034	0.043	0.049	0.052	0.054	0.056	0.056	0.057
σ_x (ft)	407	505	566	609	642	668	689	707
$\sigma_{\dot{x}}$ (fps)	82.9	88.3	91.1	92.9	94.0	94.8	95.4	95.9



(a) σ_b = 1 deg, σ_a = 32 fpss, Δt = 1 sec (b) σ_b = 1 deg, σ_a = 32 fpss, Δt = 4 sec

Figure 1. Theoretical Steady State Kalmar Filter Performance



(c) σ_b = 2 deg, σ_a = 32 fpss, Δt = 1 sec

(d) σ_b = 2 deg, σ_a = 32 fpss, Δt = 4 sec

Figure 1. - Continued.

The gain optimization process can be facilitated further, if the gains α and β are expressed by a single parameter. For this reason, we confine this analysis to the critically damped gain configuration. With this assumption, the gains are given by

$$\alpha = 1 - \gamma^2 \quad , \qquad \beta = (1 - \gamma)^2 \quad . \tag{31}$$

Here, the parameter γ is related to the filter bandwidth ω_b ($\triangleq 1/\tau_b$) by the expression

$$\gamma = \exp(-\omega_b \Delta\tau) = \exp(-\Delta\tau/\tau_b) \quad .$$

Furthermore, to account for the range effect, the position measurement error is assumed to be

$$\sigma_x \simeq r\,\sigma_b \quad . \tag{32}$$

If the expressions (31) and (32) are substituted into the rms error Eqs (25), the following are obtained

$$\tilde{x}_{rms} = \left[\frac{\gamma^4 \Delta t^4}{(1-\gamma)^4}\,\tilde{a}^2 + \frac{(1-\gamma)\,(1+4\gamma+5\gamma^2)}{(1+\gamma)^3}\,r^2\sigma_b{}^2\right]^{1/2}$$

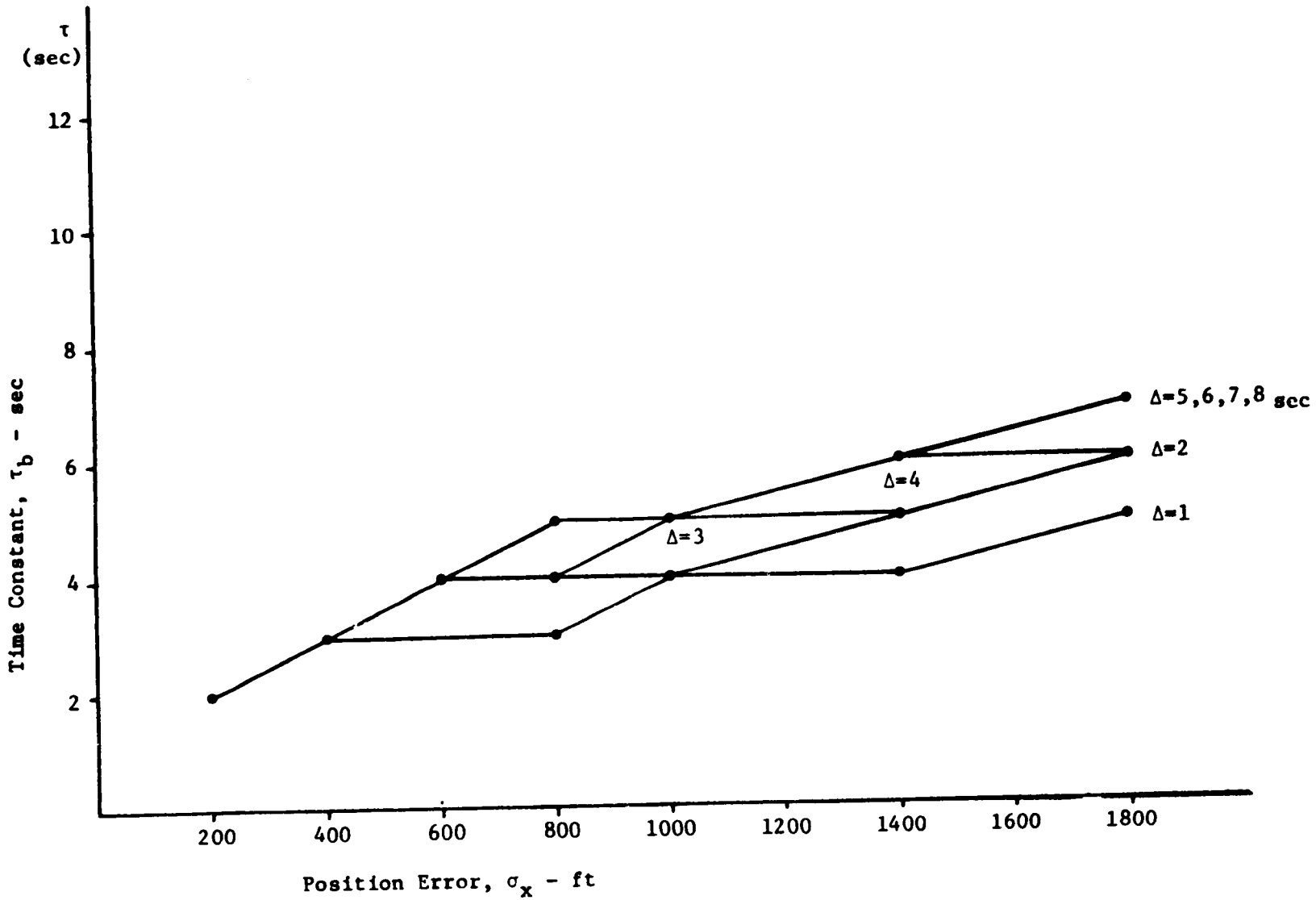
$$\dot{\tilde{x}}_{rms} = \left[\frac{(1+3\gamma)^2}{(1-\gamma)^2}\,\frac{\Delta t^2}{4}\,\tilde{a}^2 + \frac{4(1-\gamma)^3}{(1+\gamma)^3}\,\frac{1}{\Delta t^2}\,r^2\sigma_b{}^2\right]^{1/2} \tag{33}$$

The above expressions can be optimized for γ(or τ_b) in terms of the sampling period and range if proper values of $\tilde{a}$ and σ_b are given. The importance of Δt and γ is that they are filter operating parameters, whereas $\tilde{a}$ and σ_b are filter design parameters.

Figures 2a through 2c show the plots of $\tau_b{}^*$ (quantized to a nearest second) in terms of σ_x and Δt which minimizes $\dot{\tilde{x}}_{rms}$. Figure 2a corresponds to the Non-aided configuration with $\tilde{a}$ = 16 fpss; Fig 2b corresponds to Own-aided configuration with $\tilde{a}$ = 8 fpss; and Fig 2c corresponds to Own-and

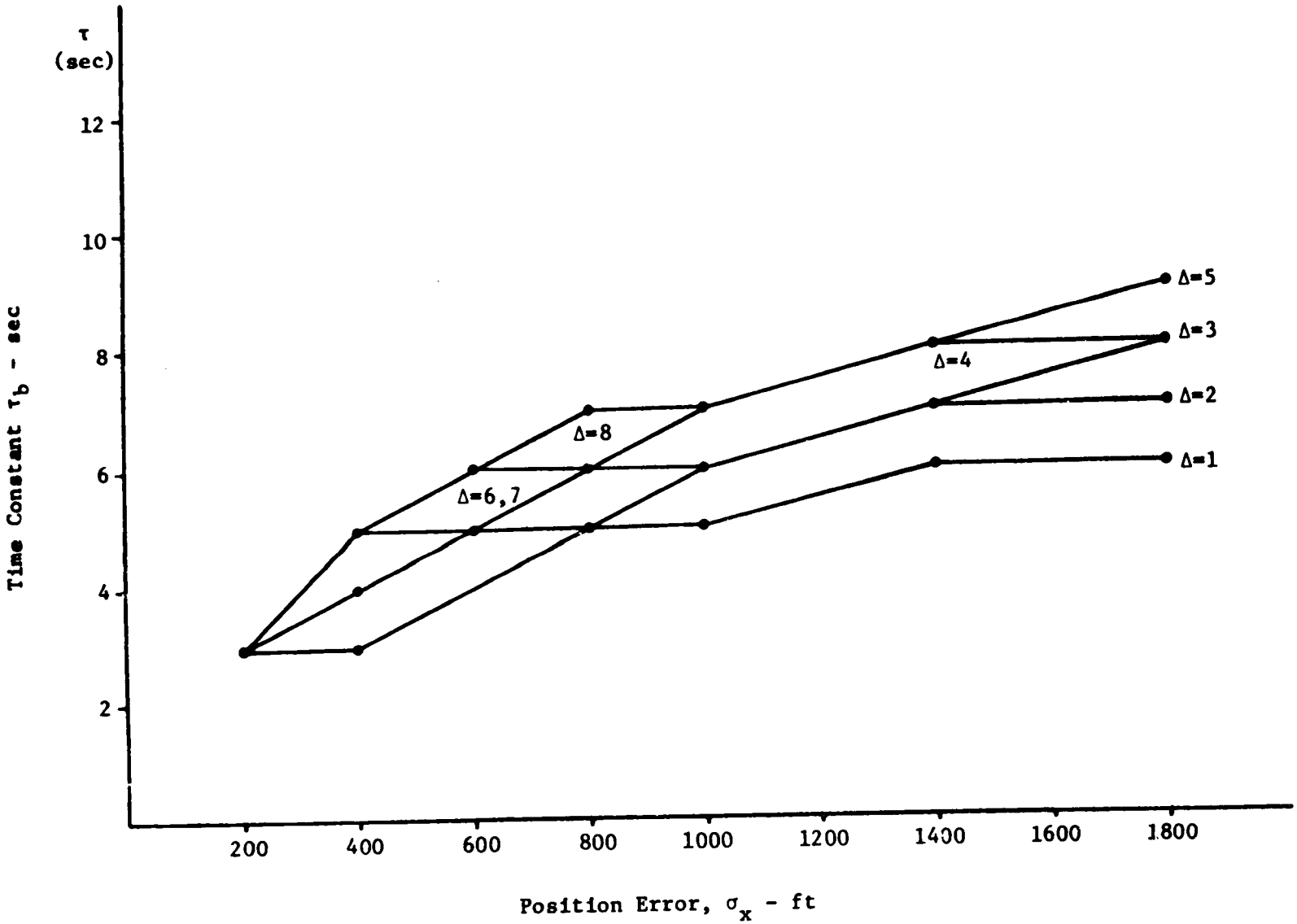
Target-aided configuration with $\tilde{a}$ = 2 fpss. When the acceleration uncertainty is high, then τ_b^* is short. When the measurement error is high, τ_b^* is long. When the sampling time is longer, then so is τ_b^*. Depending on the filter configuration, τ_b^* can be selected and interpolated with a simple two dimensional table-look-up procedure. It may be desirable to "smooth" the time constant before computing the gain values. This will prevent any occurrence of abrupt change in the filter gains.

It may turn out that this method may require as much computational load as the Kalman filter algorithm Eqs (24) through (27) when it is realistically implemented. In the next section, preliminary simulation results based on Monte Carlo passes are presented for fixed and Kalman gain filters. The results for Method (c) fall somewhere in the middle of these cases.



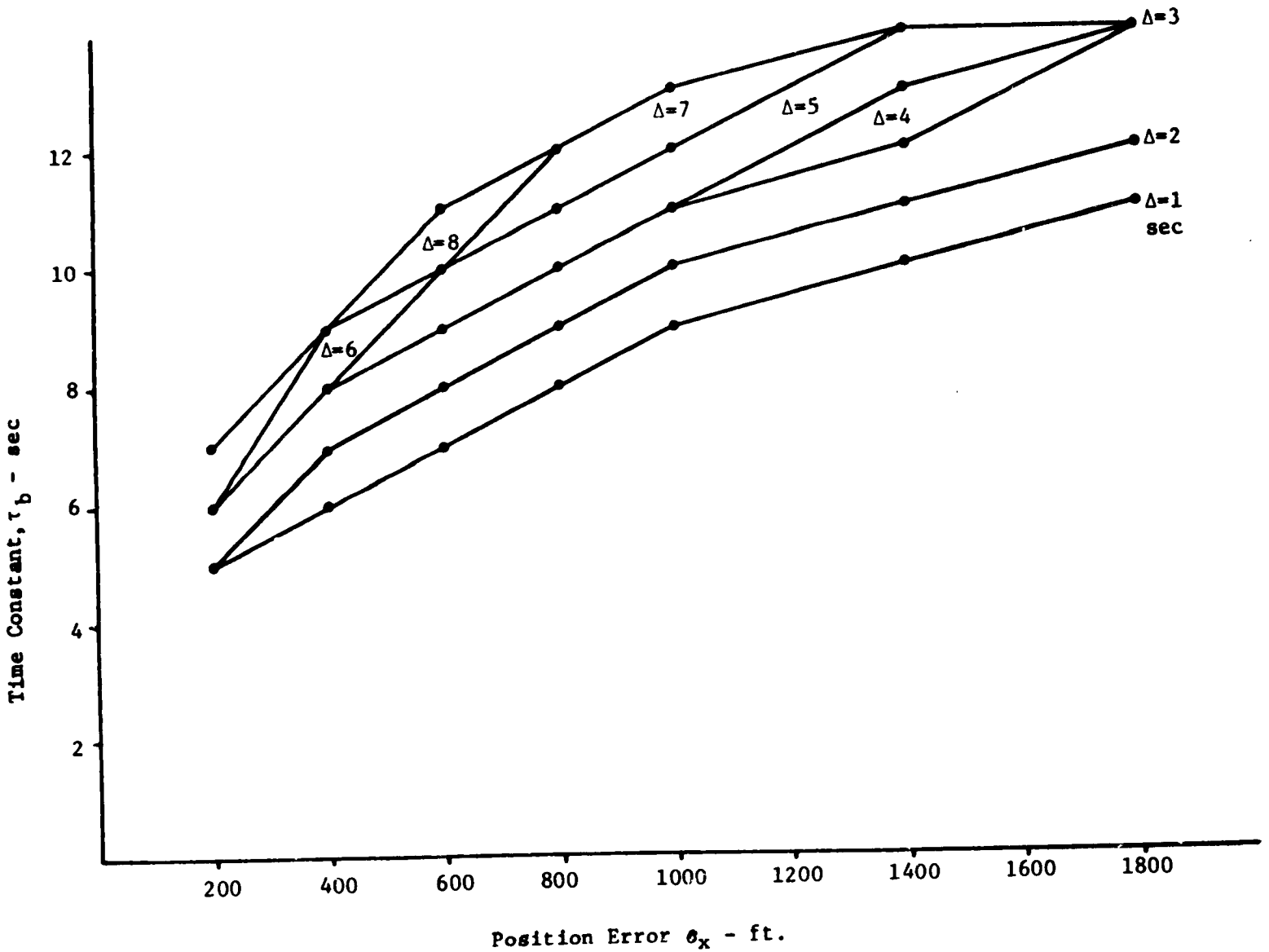
(a) Non-Aided Configuration

Figure 2. Gain Schedule



(b) Own-Aided Configuration

Figure 2. Continued



(c) Target- and Own-Aided Configuration

Figure 2. Continued

Preliminary Monte Carlo Simulation Results

A Monte Carlo simulation was set up to compare and analyze the performance of various estimation configurations. Because the impact of dynamics due to turn maneuvers is the significant key to filter performance, the various configurations are tested mostly during maneuver. Table 4 lists pertinent kinematic parameters for Own and Target aircraft. Appendix B gives a detailed description of the aircraft dynamic model used throughout the simulation study.

Table 4. Kinematic Parameters for Simulated Aircraft

	Own	Target
Ground Speed (kt)	200	382
Bank Angle (deg)	15	-15
Turn Radius (nmi)	2.2	7.2
Turn Rate (deg/sec)	1.5	-0.85

Figure 3 shows the horizontal trajectories in an earth fixed coordinate system. Figure 4 shows the relative position and velocity with respect to an Ownship north reference system, i.e., the coordinate system is not rotating as Own aircraft heading rotates. Figure 5 shows the time plot of Own and Target headings and ground speeds. Both aircraft turn simultaneously at t = 35 sec with bank commands of 15 deg (Own to the right, Target to the left). The turns last 120 sec. Because both aircraft are pulling 0.3 g, the relative acceleration could be as high as 0.6 g (20 fpss).

Original αβ Tracker Performance Figure 6 shows the statistical time plots of position and velocity errors of the TCAS αβ tracker at the sampling periods of 1, 4 and 8 seconds. The statistics are computed based on sixty (60) Monte Carlo passes and shown as the mean, mean plus one sigma and

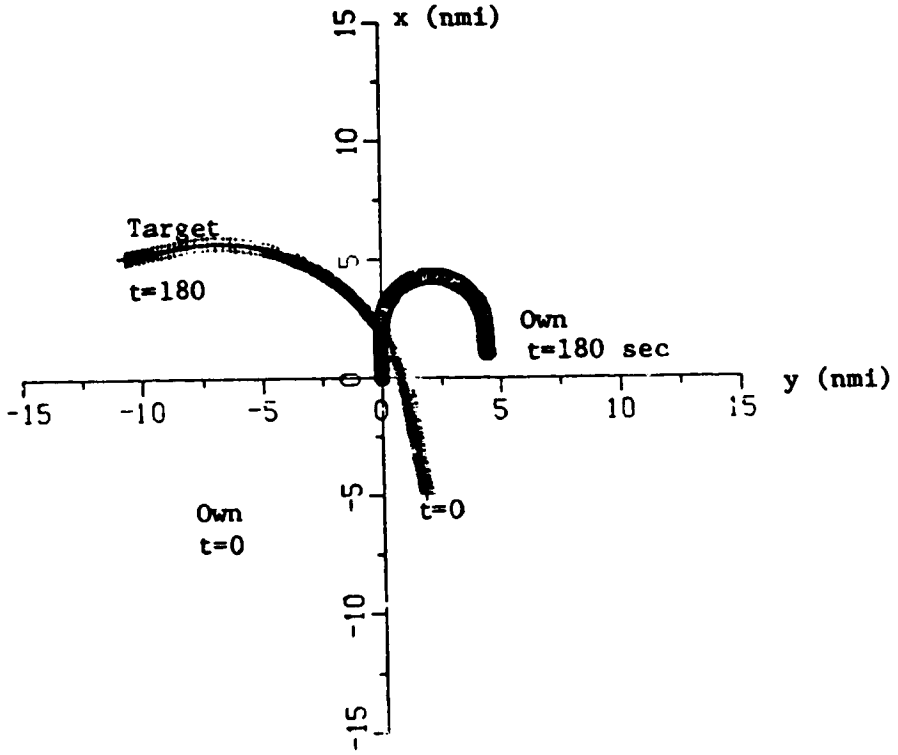


Figure 3. Own and Target Horizontal Trajectory

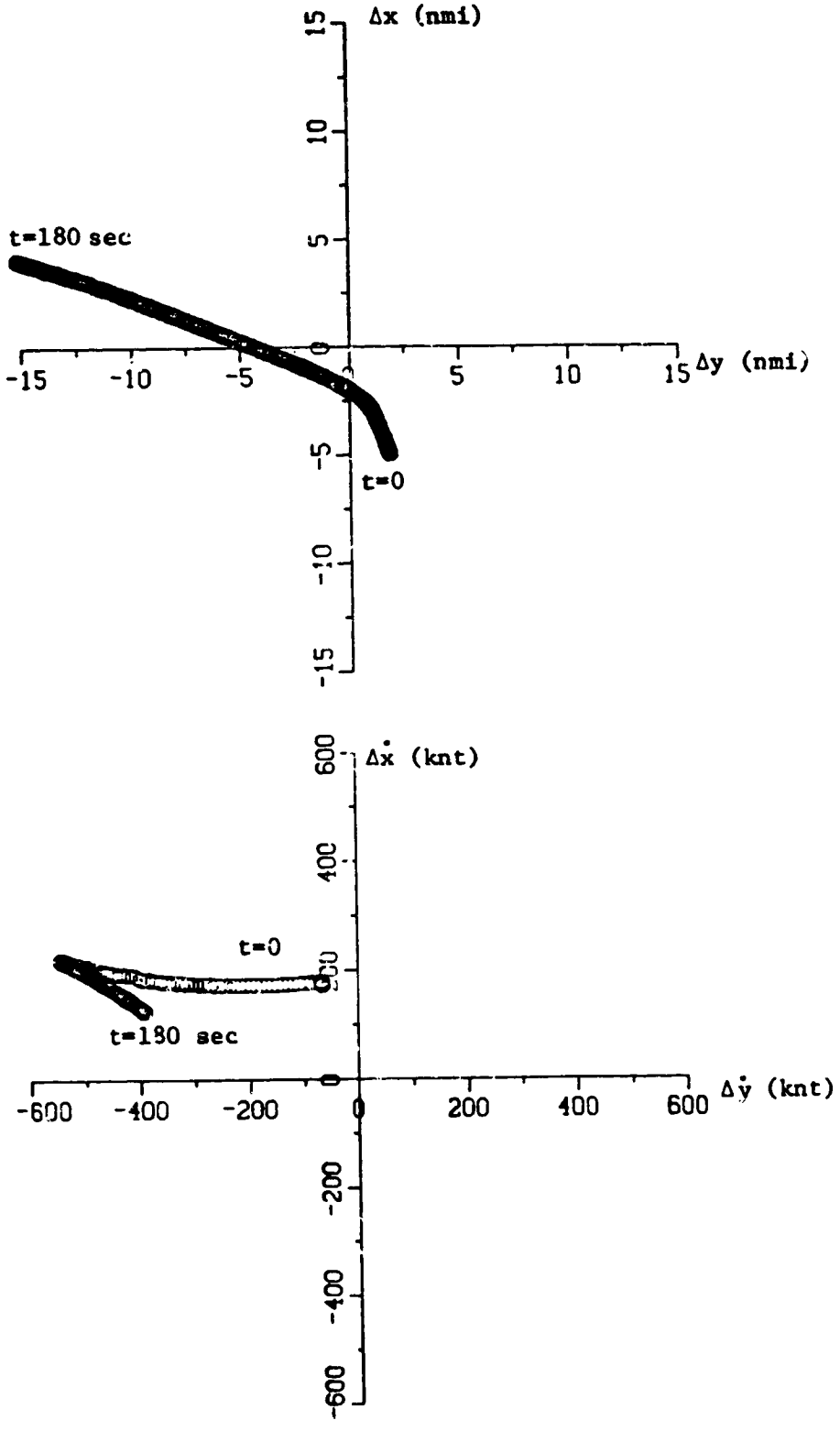


Figure 4. Relative Position and Velocity of Target with respect to Own.

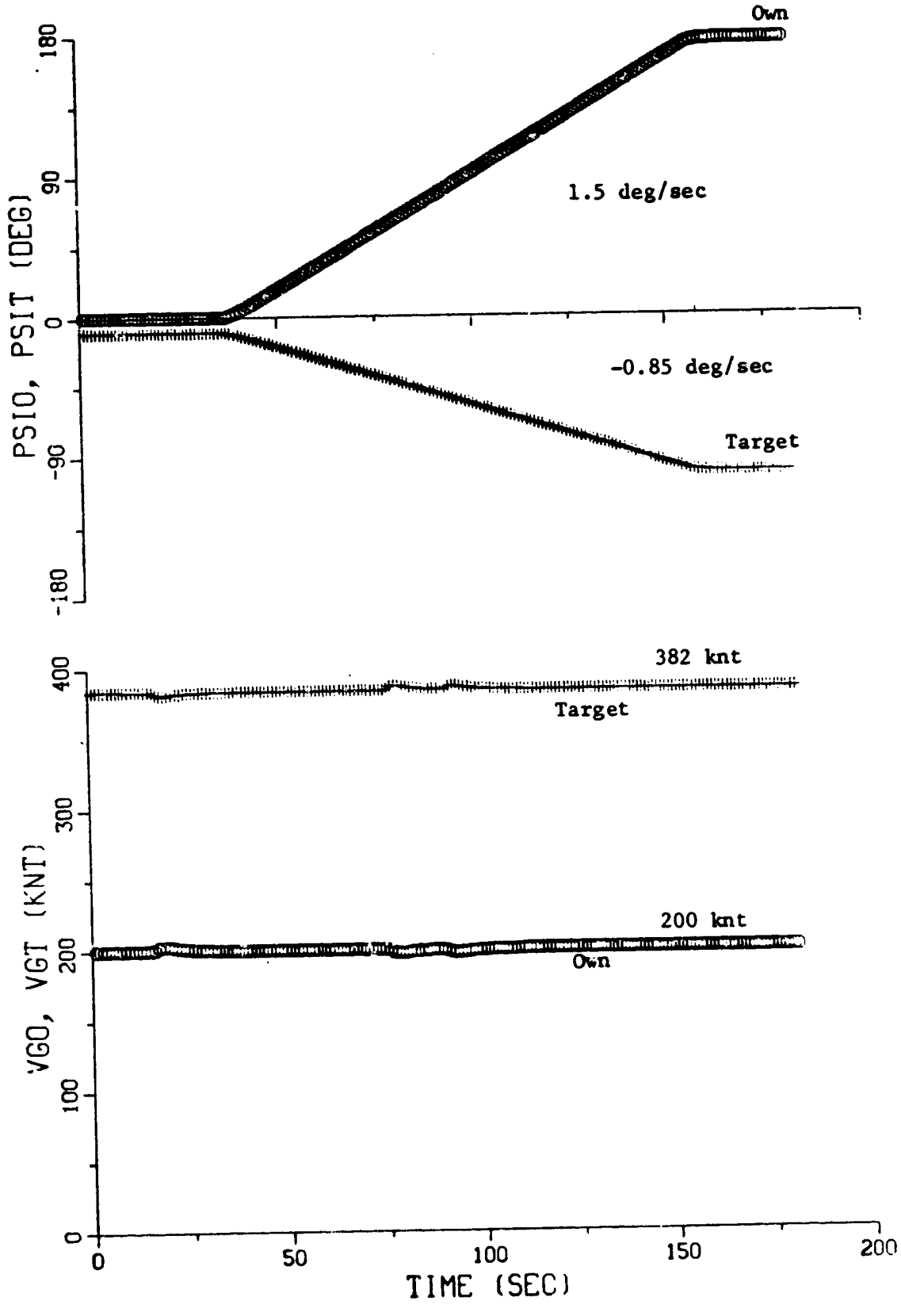
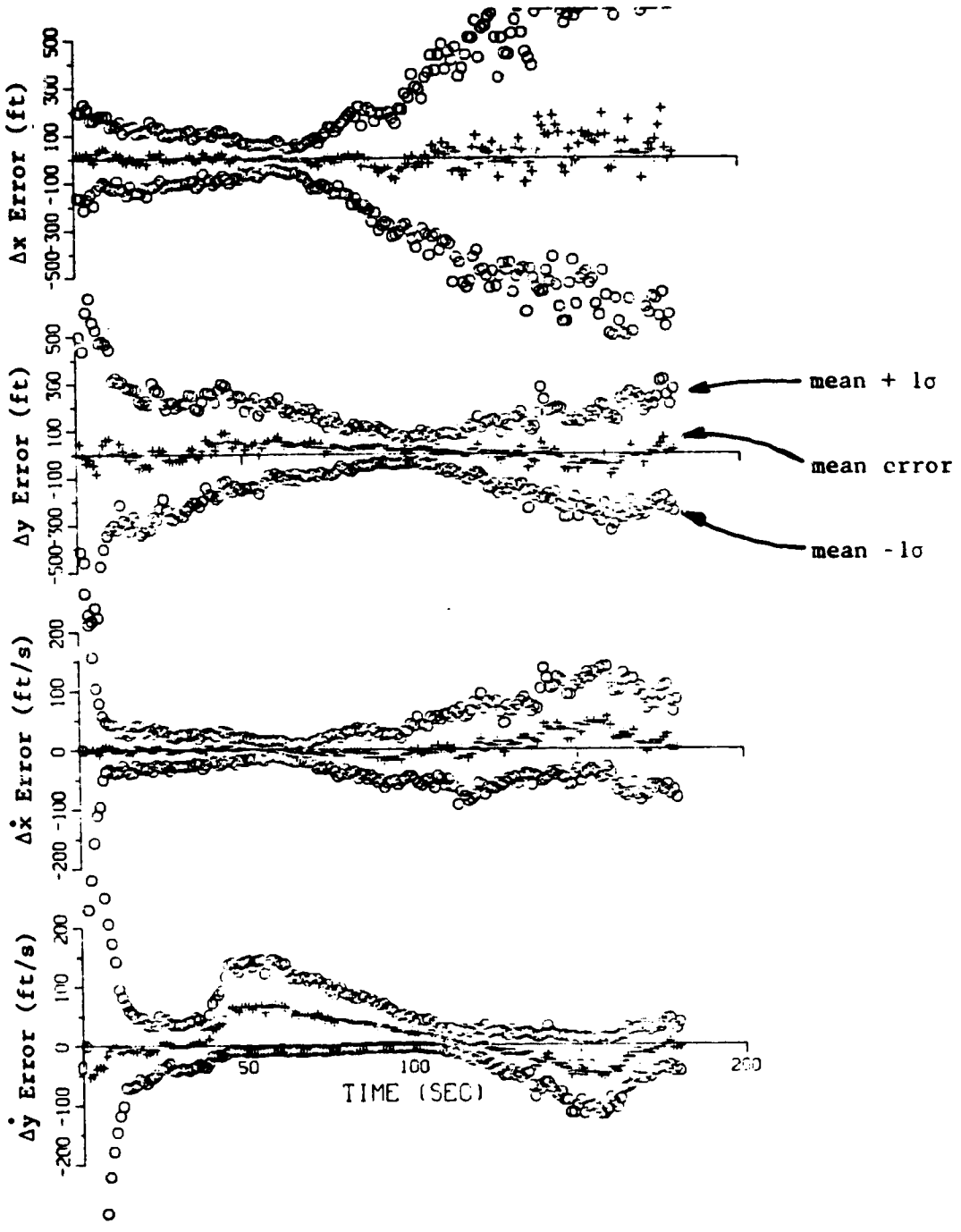
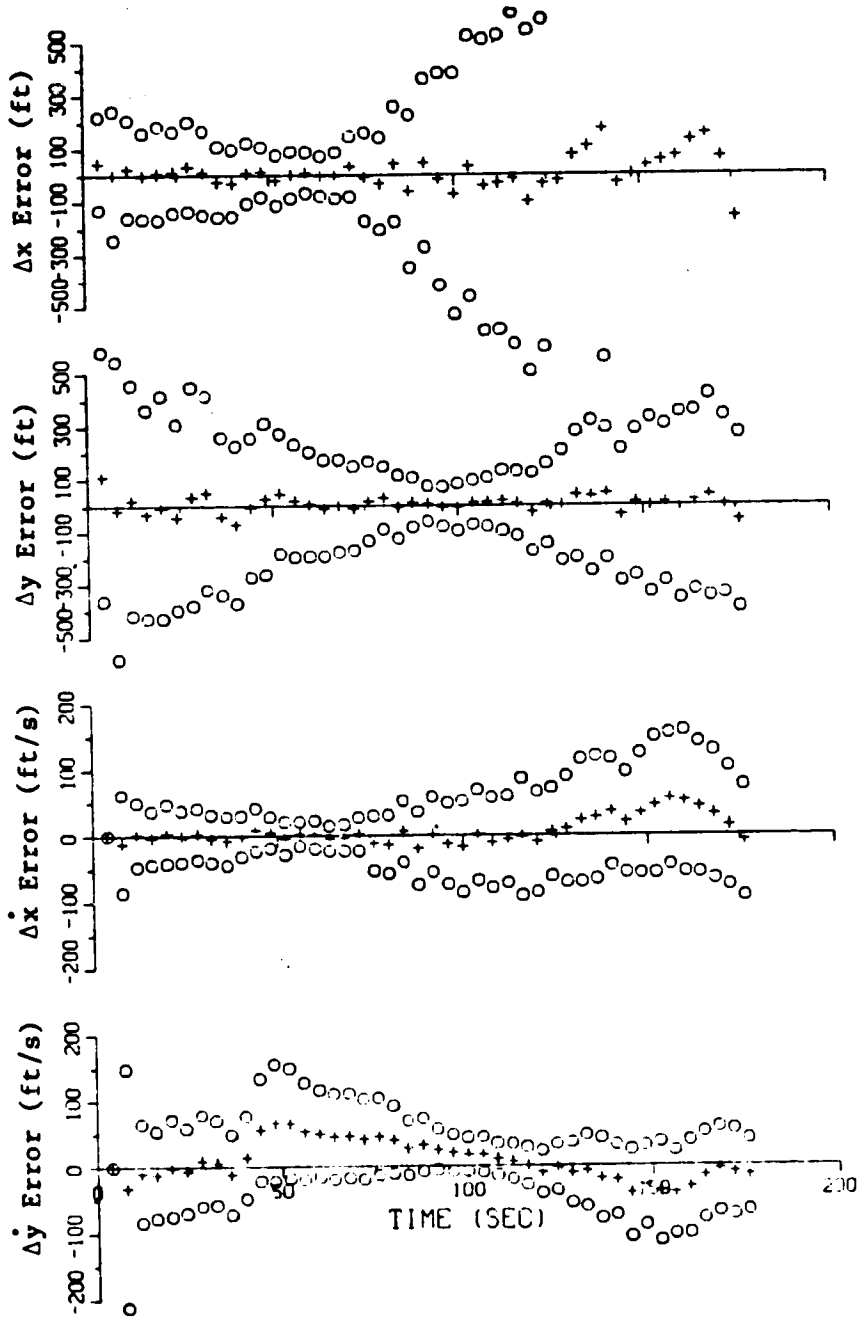


Figure 5. Time Plots of Target and Own Heading and Speed.



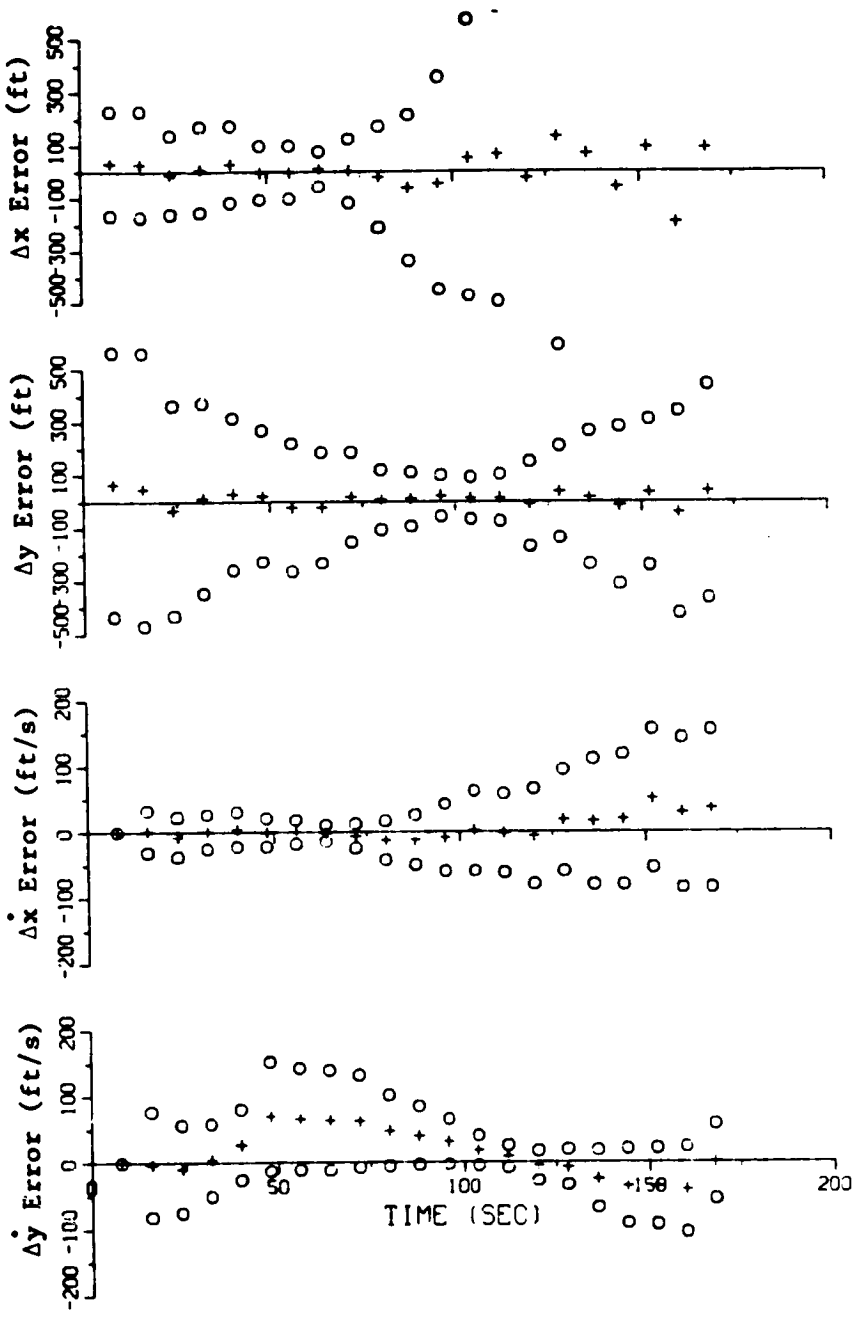
(a) Δt = 1 sec

Figure 6. Original αβ Tracker Performance



(b) $\Delta t = 4$ sec

Figure 6. Continued.



(c) Δt = 8 sec

Figure 6. Continued.

mean minus one sigma, i.e., that approximately 60% of the errors would fall between the $\pm$ sigma plots.

Some observations are listed below.

1) The range effect is clearly shown. As the target closes in range, the errors get smaller; as the target flies away, the errors get larger.

2) The mean position errors remain relatively small (within 250 ft). The dynamic delay effect of turn maneuvers does not seem to be significant. This implies that the position gains are on the high side. The maximum rms error is approximately 1000 ft at 16 nmi range. Position errors tend to increase as the sampling time increases. (This was predicted by the theoretical results.)

3) The velocity errors show the same range effect characteristics. The mean errors show the marked dynamic delay error especially for the y-axis. It is apparent from Fig.6 that the y-axis contains most of the dynamics. The velocity gains are well tuned in the sense that the dynamic error and the high frequency error have similar magnitude. The maximum rms error reaches approximately 100 kt. Except for the initial transients, the error curves are similar. It is noted that the initial velocity estimate using the first two consecutive measurements becomes more accurate as the sampling period becomes longer because of the greater signal-to-noise ratio.

Effect of Limitors on the Feedback Signal In many cases of navigation filter or feedback regulator control system design, a limitor is used on the feedback signal to prevent "unreasonable" error signals disturbing the system. In equation form, a limitor is given by

$$\tilde{\Delta x} \leftarrow \begin{cases} \tilde{\Delta x} & , \text{ if } |\tilde{\Delta x}| \leq L_x \\ \text{sign } (\tilde{\Delta x})\, L_x & , \text{ if } |\tilde{\Delta x}| > L_x \end{cases} . \tag{34}$$

That is, the feedback signal $\tilde{\Delta x}$ may be replaced by the term L_x in the filter Eq (8) depending on the magnitude. The y signal is generated in a similar manner.

Figure 7(a) shows the Monte Carlo simulation results with

$$L_x = \sigma_x \quad \text{and} \quad L_y = \sigma_y \tag{35}$$

at the sampling period of 1 sec. While the x-axis behavior remains "reasonable", the y-axis clearly shows the unstable nature mostly due to not being able to track velocity. Figure 7(b) shows the case when two sigma values are used as the limitor value at Δt of 5 sec. In this case, the unstable nature becomes apparent at Δt of 4 sec or longer (Δt of 5 sec is shown).

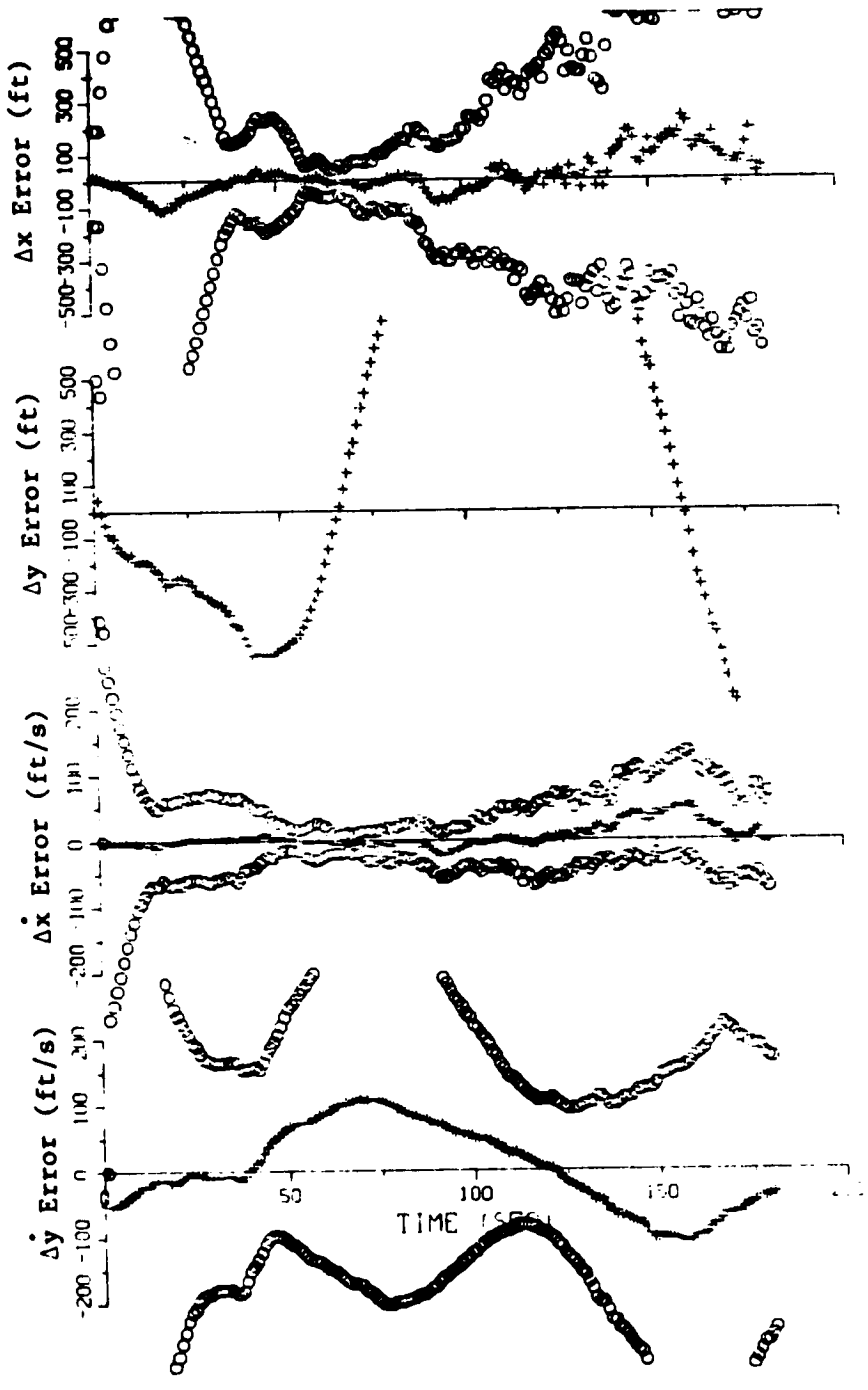
Other nonlinear devices of this nature were tested with similar results. For example, a soft limitor defined by a quadratic function (rather than a straight line linear function) was also tested. This indicates that the performance of the current tracker can not be improved very much with these devices.

Kalman Filter Algorithm Gains The Monte Carlo Simulation program was modified to include three filter configurations. These are Non-aided, Own-aided and Target-and Own-aided configurations. Instead of fixed gains, the filters utilize the Kalman filter algorithm for computing gains (Eqs (27) through (29)). Figures 8 and 9 show the statistical error time plots of three filters side by side for the sampling period of 1 and 4 seconds. Figure 8 shows the case with the gains set at high values and Fig. 9 shows the case with the gains set at low values.

The gain computations are done automatically. But, by adjusting the acceleration uncertainty value, σ_a , high or low gain can be selected. The input acceleration uncertainties are shown in the following Table 5.

Table 5. Input Acceleration Uncertainties
($\sigma_{\ddot{x}}$ or $\sigma_{\ddot{y}}$ in f/s^2)

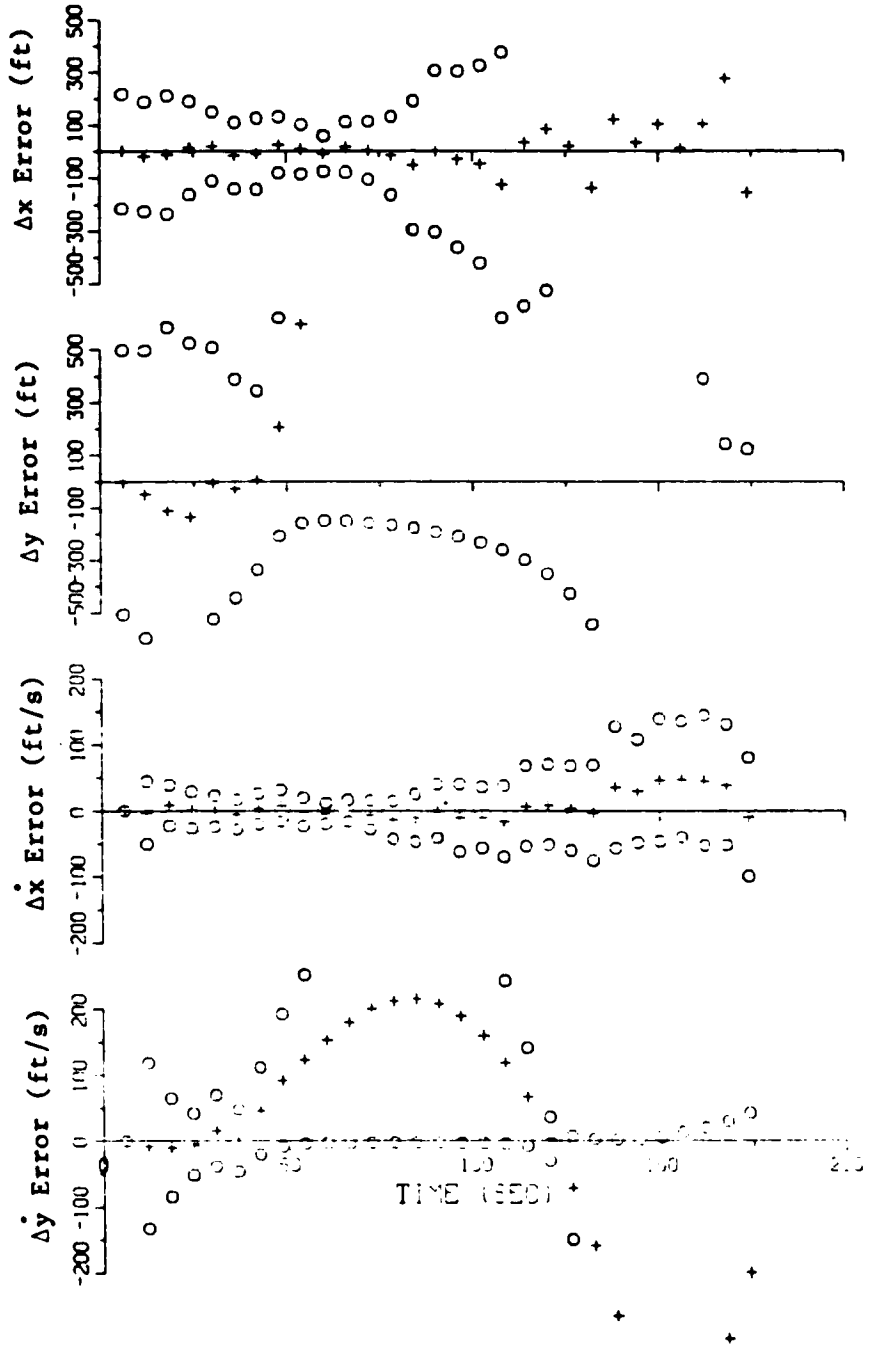
	Non-Aided	Own-Aided	Target- and Own-Aided
High Gain	32	32	32
Low Gain	20	16	8



(a) Limit = one sigma of measurement error

Δt = 1 sec

Figure 7. Effect of Limitor on the Feedback Signal



(b) Limit = two sigma of measurement error

Δt = 5 sec

Figure 7. Continued

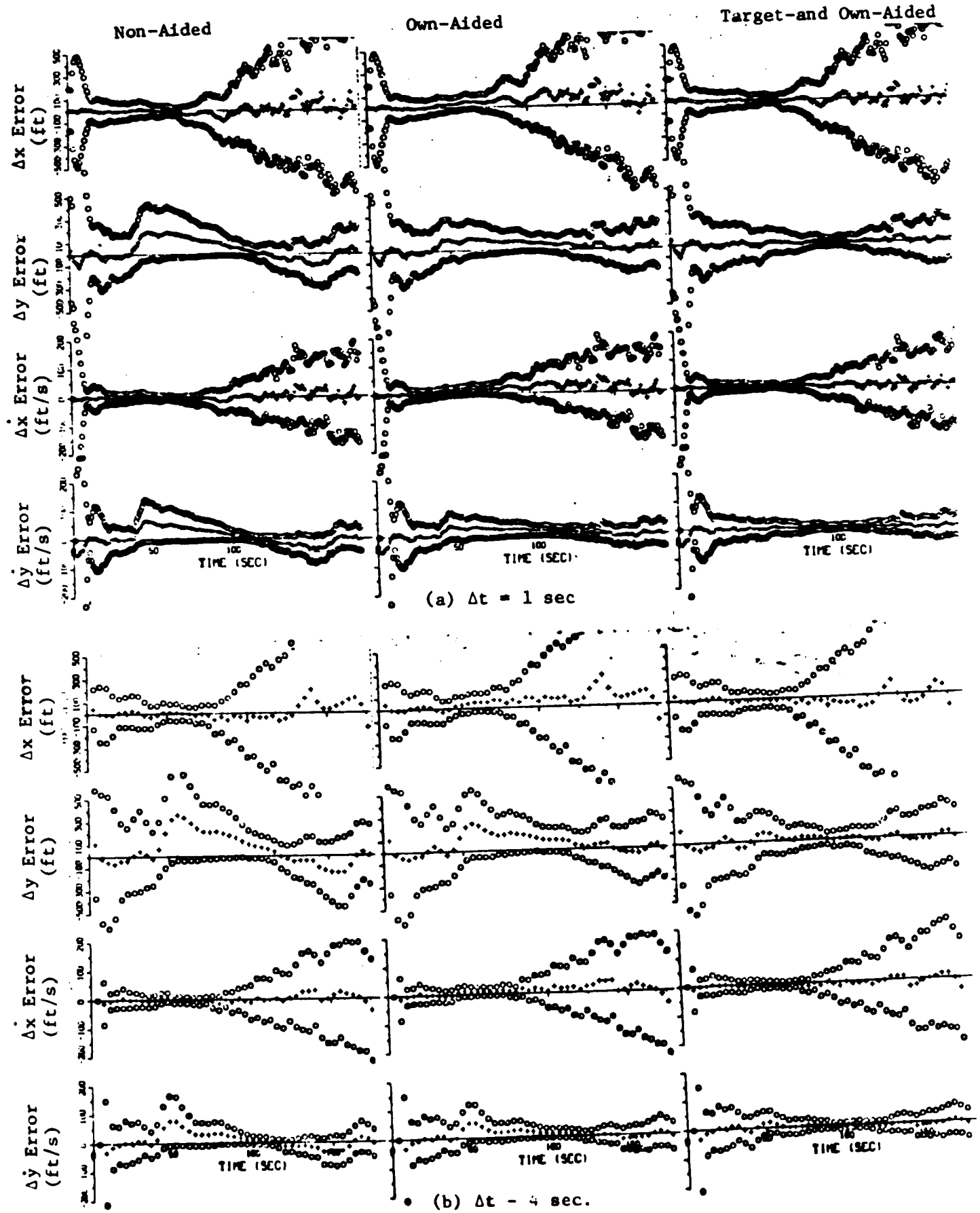
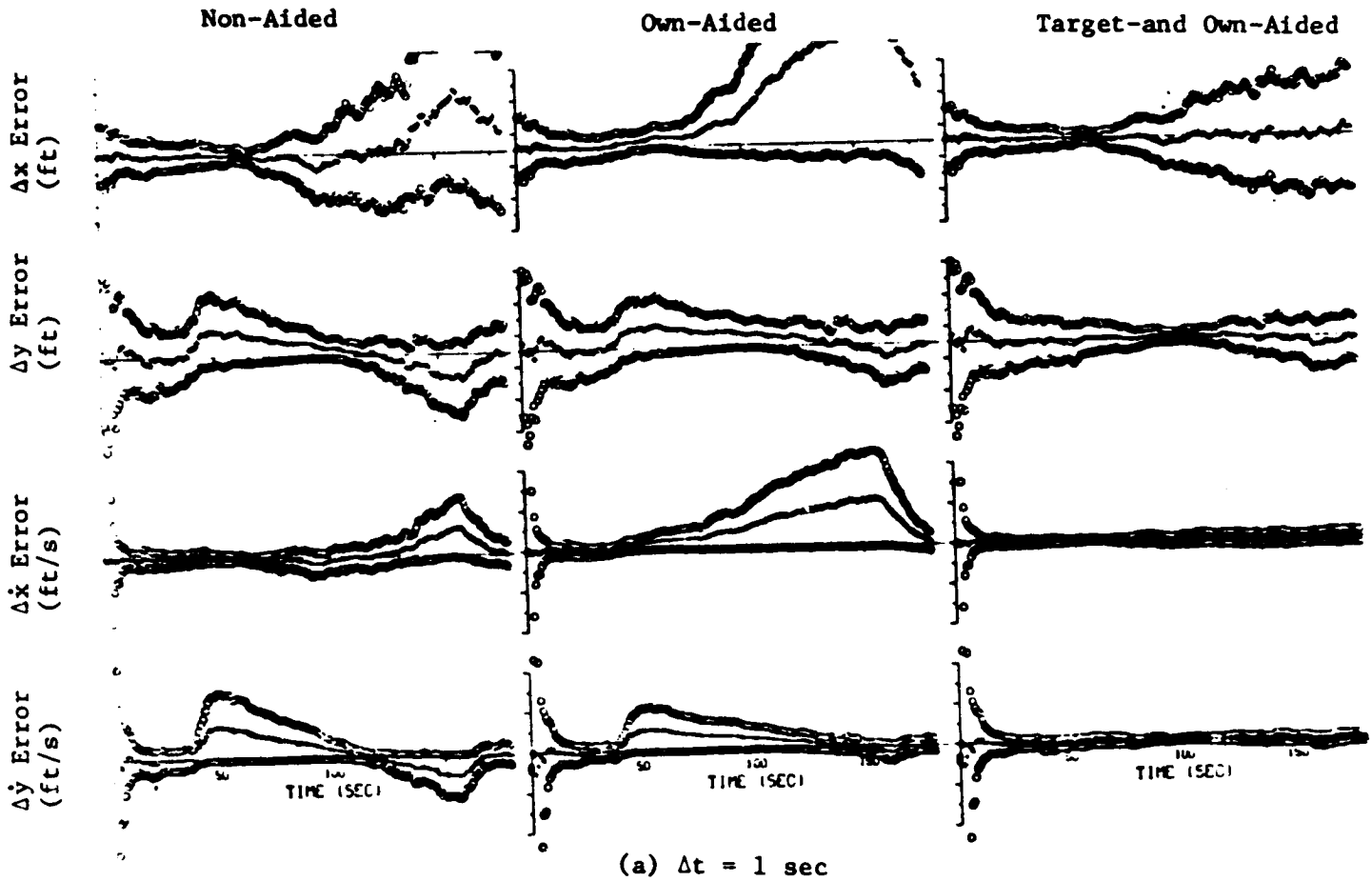
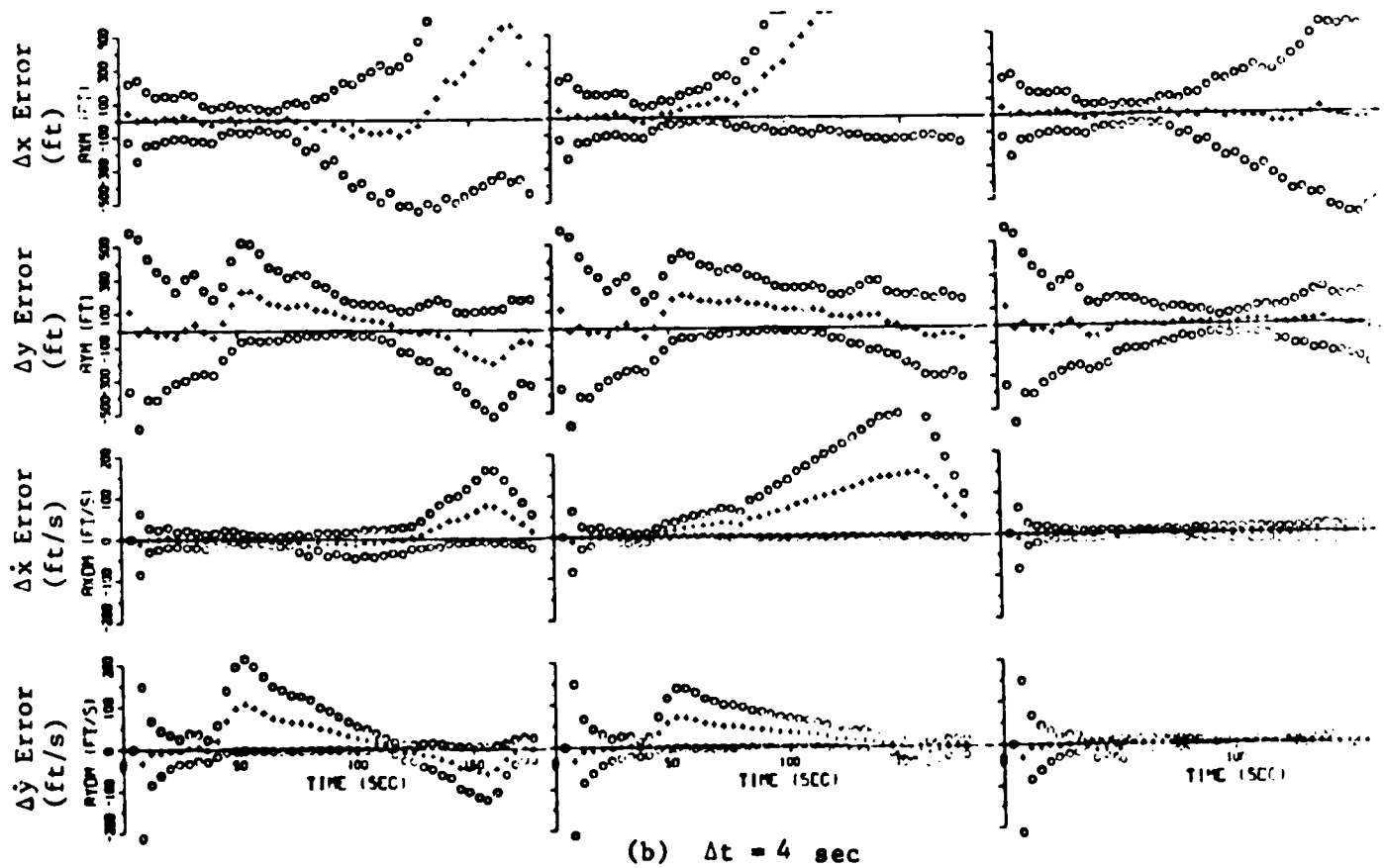


Figure 8. High Kalman Filter Gains for Three Filter Configurations.



(a) Δt = 1 sec



(b) Δt = 4 sec

Figure 9. Low Kalman Filter Gains for Three Filter Configurations.

The rationale of using successively lower σ_a's for the low filter gains is that the uncertainty should decrease as Own and Target onboard signals are used to complement the relative position measurement. Note that for the high gain case, the acceleration uncertainty value is the same for all three configurations. Thus, the gains would be identical. This implies that the smoothing characteristics of alternating the measurement noise magnitude would be the same.

Now referring to Fig. 8, the $\pm$ one sigma envelopes are very similiar for all these configurations confirming the intuition expressed in the last paragraph. Even though the gains are computed to attenuate the range effect by lowering the gains at longer range, the effect is still very much prominent. The performance of the x-axis of the non-aided configuration looks similar to a fixed gain case. The y-axis shows similar traits; however, the mean position and velocity errors are larger for the Kalman filter case. This points out that x- and y-axis filter gains are computed separately. This means that the Kalman filter band-width for the y-axis is narrower than the constant gain filter. The mean $\dot{y}$ error for the 4 sec case shows smaller value.

For this particular case of relative kinematics (a counter example will be shown later), the Own data and Target and Own data complementation successfully improve the tracking performance. This can be seen as smaller mean errors. For the case of Target and Own data complementation, the mean errors are remarkably close to zero even for the 4 second sampling time case.

Referring to Fig. 9, the following observations can be made :

(i) Performance of the x-axis for both Non-aided and Own-aided configurations deteriorates. The major cause for this is the larger mean errors due to dynamic delay. Performance of the Own-aided configuration is larger than that for the Non-aided configuration. For the y-axis, the performance of both configurations are very similar. These comments are applicable for $\Delta t = 4$ sec case also.

(ii) For the Target- and Own-aided configuration, errors are generally smaller. Also, the mean errors are small regardless of sampling time. The range effects are very slight in the velocity errors. This seems to indicate that the position measurements are used mainly to update the low frequency error in the average acceleration input and the high frequency input error is "integrated" out.

Non-Aided and Own-Aided Configurations It could be concluded that a partial indication of maneuvers is better than none. In particular, it seems true that complementing with Own data, even though lacking Target data, is better than no complementation at all. This would be true in general cases of the relative kinematics involving only Own aircraft maneuvering. In the case of both aircraft maneuvering, the filter performance depends on relative acceleration. For example, in the example used in the previous sections, the Own-aided filter performed better than the non-aided filter if the filter gains are the same.

A counter example can be given wherein this is not the case in that instead of continuing to turn at the given bank angles, both aircraft go into bank reversal maneuvers simultaneously at the mid-point. Figures 10 through 12 show the pertinent parameters. It is noted from the relative velocity plot of Fig. 11, that the relative x velocity is nearly constant. Therefore, in this case, the non-aided configuration should perform better. Figure 13 shows the error time plots of the Monte Carlo run. Both filters are set in the high gain Kalman filter mode. As can be seen, the position errors are comparable. The relative x-velocity estimates are markedly different : The non-aided filter shows a small mean error; whereas, the Own-aided filter shows a substantial non-zero mean error. The fact that the standard deviations are similar in size indicates that the differences in characteristics are due solely to the difference in (absence or presence of) acceleration input.

Conclusions

Important conclusions are summarized below:

(1) Combination of Target-and Own-data complementation and the Kalman filter gain computation exhibited the best results. One advantage is that the errors due to the dynamic lag induced by maneuvers are non-existent;

(2) Own-data complementation helps in cases where only Own is maneuvering. If both Own and target are maneuvering, then the estimation performance depends on the relative acceleration;

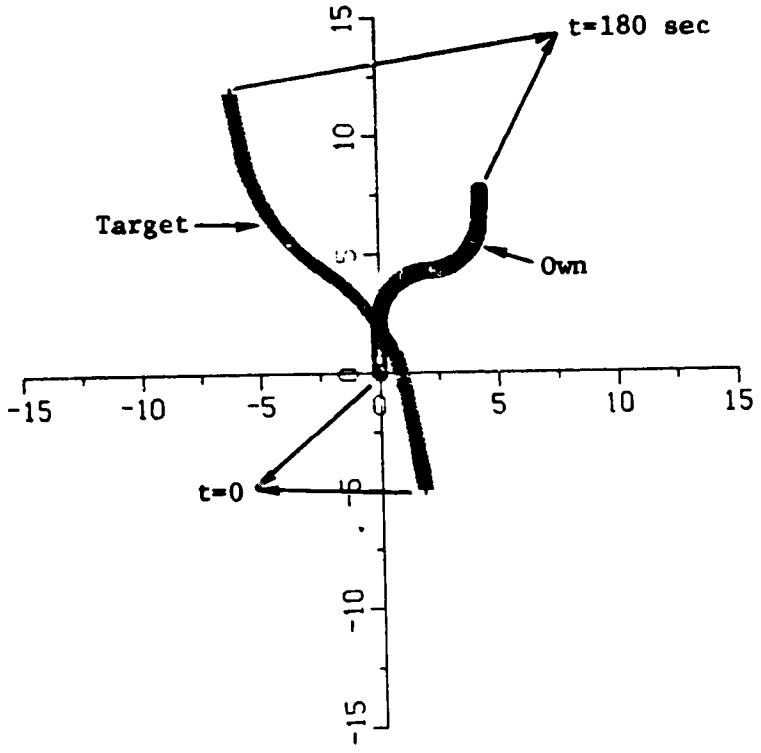


Figure 10. Own and Target Horizontal Trajectory

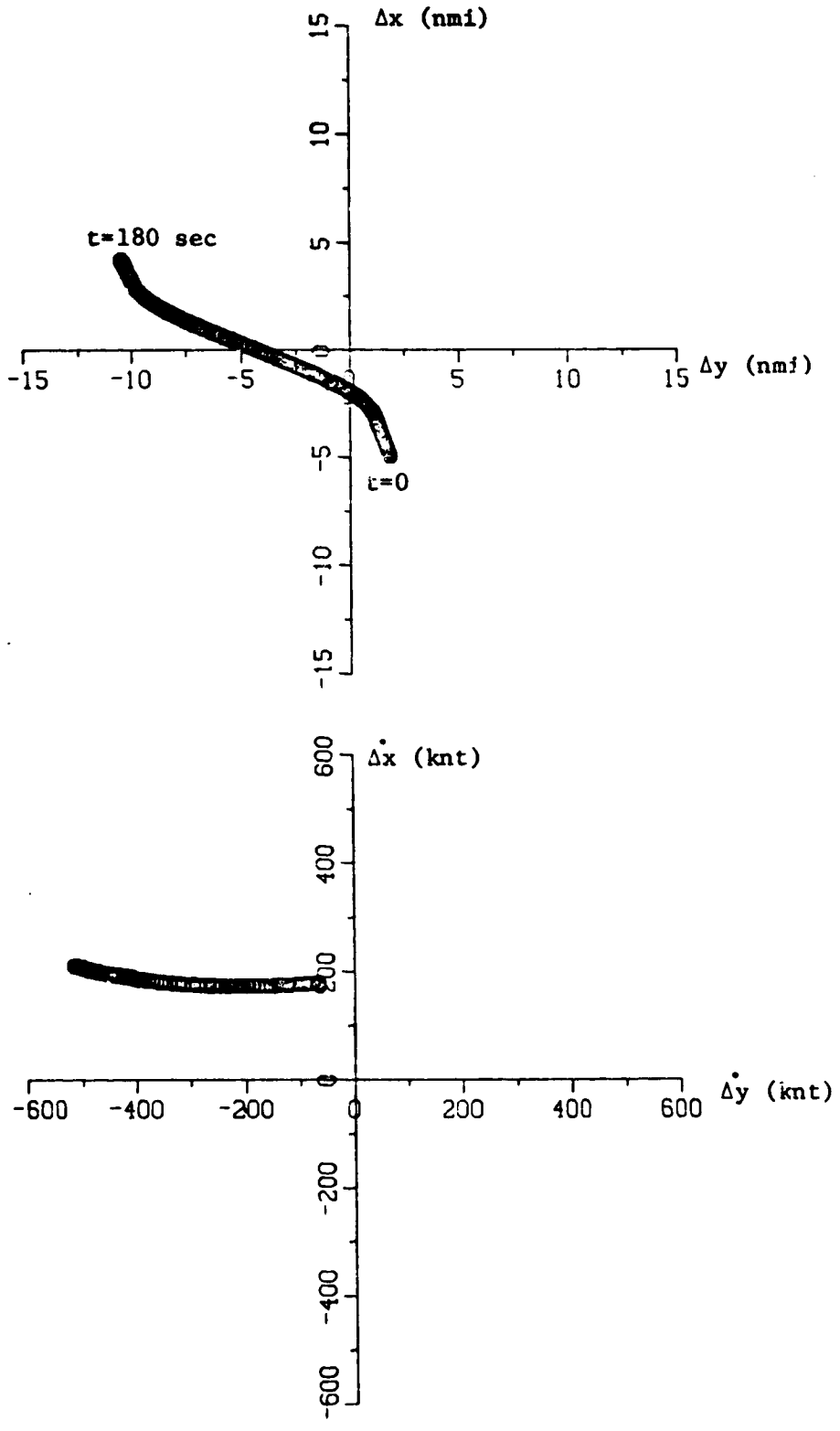


Figure 11 Relative Position and Velocity with respect to Own

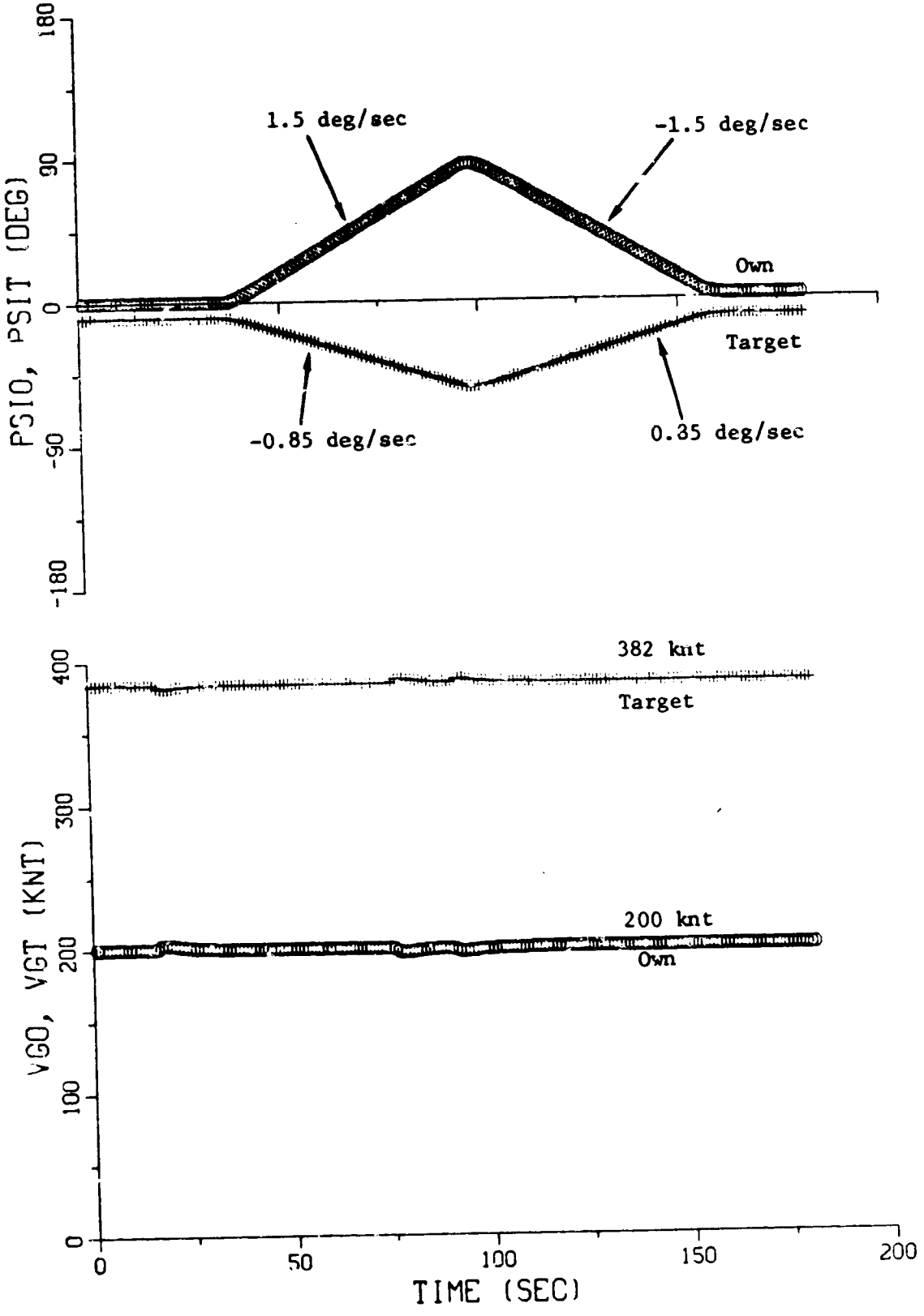


Figure 12. Time Plots of Target and Own Heading and Speed

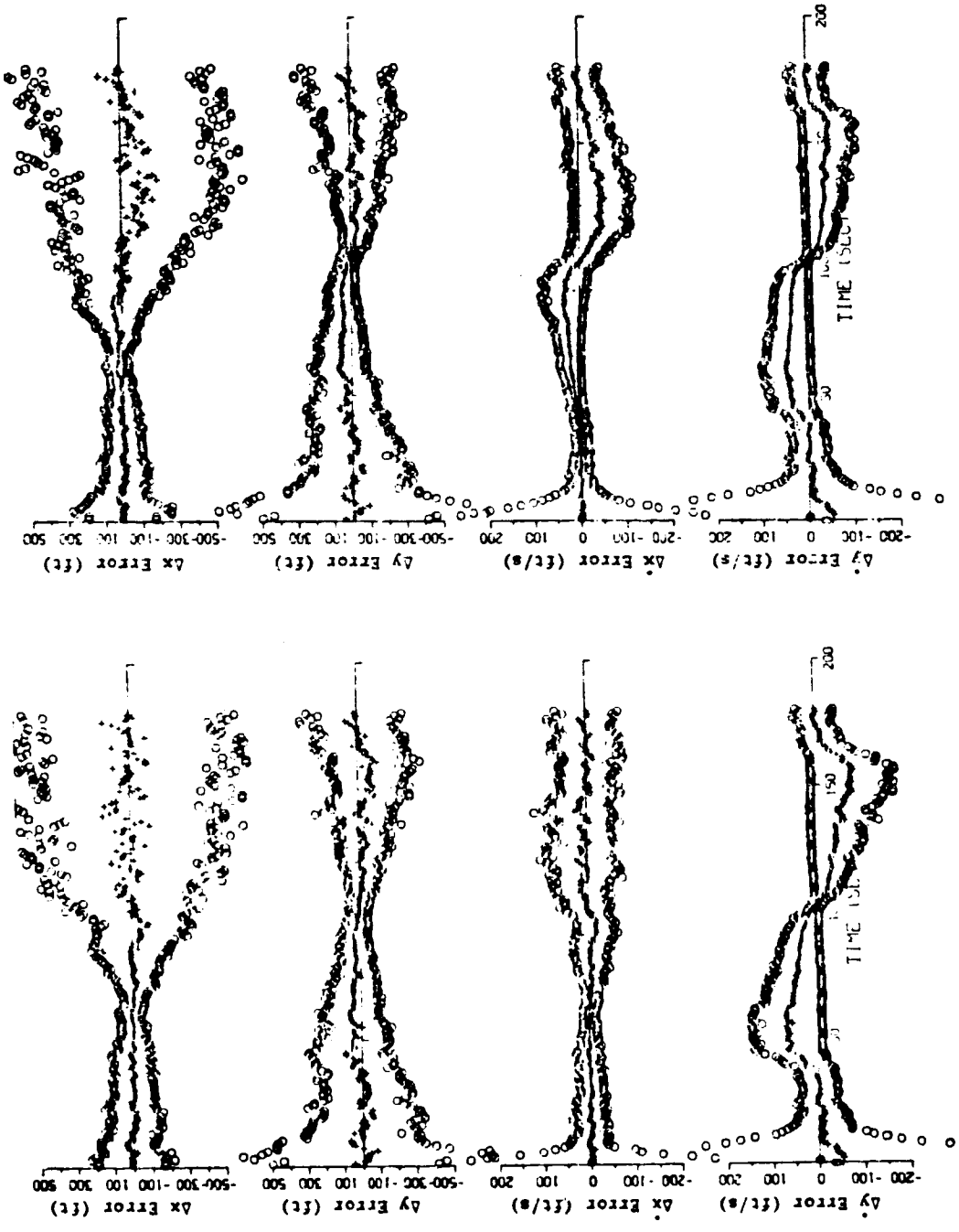


Figure 13. Statistical Time Plots of Position and Velocity Estimation Errors for Non-Aided and Own-Aided Filters.

(3) Non-aided configuration with the constant gains has problems when either aircraft undergoes a maneuver. However, as will be shown later, its performance is credible when the relative kinematics is rectilinear. Because the Kalman filter gain computations do not require significant real time, this method rather than fixed gains is recommended; and

(4) Adaptive feature of monitoring the innovations (or measurement residues) to detect dynamic delay error was not considered. This feature may be necessary to avoid using estimates which contain large and sustained dynamic lag errors. Without such a feature, the system does not know "when it does not know".

The impact of these errors is discussed in Chapter V with respect to CDTI and CAS applications. Also in the next chapter, their impact is discussed with respect to the range and range rate estimate derivations.

III

RANGE FILTERS

Introduction

By far the two most important kinematic variables for the TCAS system are relative range and altitude. These are monitored very carefully for targets in the vicinity of Own aircraft to determine separation status. The monitoring process involves essentially three steps: measurement, filtering and collision avoidance logic. The filtering algorithm provides the dynamic state estimates to the collision avoidance logic based on the measurements obtained by the surveillance subsystem.

Note: The planned TCAS II software architecture incorporates similar filter algorithms in two different submodules, i.e., the filter algorithms are not separate from the other two functions. One set of filters is used within the surveillance subsystem. It is used to correlate and distinguish the internal track files and incoming measurements. The so-called "gating" techniques are used for this purpose. The gate's upper and lower thresholds are dynamically generated based on the predicted state variables.

Another set of filtering algorithms are incorporated within the collision avoidance logic. Of course, these are used to provide the threat assessment. In our discussion, the filtering algorithms are treated as an independent functional block.

For the range axis, the draft TCAS II Minimum Operational Performance Standards [9] proposes three filtering methods:

(1) the horizontal x-y filter ;

(2) the range $\alpha\beta$ tracker ; and

(3) the range squared $\alpha\beta\gamma$ tracker.

These are discussed in the following sections.

Range Filter Algorithms

Range and Range Rate Estimates Derived from Horizontal Δx-Δy Estimate This method is recommended in the draft TCAS II MOPS when reasonably accurate relative bearing is obtainable. The algorithm is simple. The relative range Δr is given in terms of the Δx-Δy horizontal components by the formula

$$\Delta r = [\Delta x^2 + \Delta y^2]^{1/2} \quad . \tag{36}$$

Taking time derivative

$$\begin{aligned} \dot{\Delta r} &= [\Delta x^2 + \Delta y^2]^{-1/2} (\Delta x \cdot \dot{\Delta x} + \Delta y \cdot \dot{\Delta y}) \\ &= (\Delta x \cdot \dot{\Delta x} + \Delta y \cdot \dot{\Delta y})/\Delta r \quad . \end{aligned} \tag{37}$$

The range and range rate estimates can be obtained by substituting the horizontal estimates obtained in Chapter II.

The range and range rate errors are related to the horizontal x-y errors (within the linear term) by

$$\begin{aligned} \tilde{\Delta r} &\doteq \text{cosb}\ \tilde{\Delta x} + \text{sinb}\ \tilde{\Delta y} \quad , \\ \dot{\tilde{\Delta r}} &\doteq \frac{1}{\Delta r} \{\text{sinb}\ [\text{sinb}\ \dot{\Delta x} - \text{cosb}\ \dot{\Delta y}]\tilde{\Delta x} \\ &\qquad + \text{cosb}\ [\text{cosb}\ \dot{\Delta y} - \text{sinb}\ \dot{\Delta x}]\tilde{\Delta y}\} \\ &\quad + \text{cosb}\ \dot{\tilde{\Delta x}} + \text{sinb}\ \dot{\tilde{\Delta y}} \quad , \end{aligned} \tag{38}$$

where

$$\text{cosb} = \frac{\Delta x}{\Delta r} \quad , \quad \text{sinb} = \frac{\Delta y}{\Delta r} \quad . \tag{39}$$

Equations (38) simply state that the range and range rate errors are the same order of magnitude as the respective horizontal components.

The Range $\alpha\beta$ Tracker In this configuration, the measured range is fed into an $\alpha\beta$ tracker algorithm. The filtering algorithm is repeated below.

$$
\begin{aligned}
&\text{range prediction:} && \Delta r^{+} = \Delta \hat{r}_n + \Delta t \cdot \dot{\hat{\Delta r}}_n \\
&\text{range error:} && \tilde{\Delta r} = \Delta r^{m}_{n+1} - \Delta r^{+} \\
&\text{range update:} && \hat{\Delta r}_{n+1} = \Delta r^{+} + \alpha\, \tilde{\Delta r} \\
&\text{range rate update:} && \dot{\hat{\Delta r}}_{n+1} = \dot{\hat{\Delta r}}_n + (\beta/\Delta t)\tilde{\Delta r} \quad .
\end{aligned}
\tag{40}
$$

When the range measurement is missing, the predicted range is used for a short duration of time (6 sec) before the surveillance process is re-started again. The first two consecutive measurements are used to initialize the states. The recommended α and β gains are 0.67 and 0.25 respectively for the nominal sampling interval of 1 sec.

The Range-Square $\alpha\beta\gamma$ Tracker One of the major drawbacks of the range $\alpha\beta$ tracker is that the range is a nonlinear function of time even though the underlying relative kinematics is rectilinear. For example, if

$$\Delta x = \Delta x_0 + \dot{\Delta x} \cdot t \quad ; \quad \Delta y = \Delta y_0 + \dot{\Delta y} \cdot t$$

with constant $\dot{\Delta x}$ and $\dot{\Delta y}$, then the range is given by

$$\Delta r = \left[(\Delta x_0^{\,2} + \Delta y_0^{\,2}) + 2(\Delta x_0\, \dot{\Delta x} + \Delta y_0\, \dot{\Delta y})t + (\dot{\Delta x}^2 + \dot{\Delta y}^2)t^2 \right]^{1/2} \tag{41}$$

The above expression can be approximated by the linear expression

$$\Delta r \approx \Delta r_0 + \dot{\Delta r}\, t \quad ,$$

which forms the basis for the $\alpha\beta$ tracker formulation, if $\dot{\Delta r}$ is reasonably constant. See Eq (40).

Equation (41) forms the basis of the range square filter. By squaring both sides of Eq (41), one obtains a quadratic equation

$$s \equiv \Delta r^2 = s_0 + \dot{s}_0 t + \frac{1}{2}\ddot{s}_0 t^2 \quad , \tag{42}$$

where

$$s_0 = \Delta x_0^2 + \Delta y_0^2 ,$$
$$\dot{s}_0 = 2(\Delta x_0 \dot{\Delta x} + \Delta y_0 \dot{\Delta y}) ,$$
$$\ddot{s}_0 = (\dot{\Delta x}^2 + \dot{\Delta y}^2) .$$

The quadratic coefficients s_0, $\dot{s}_0$ and $\ddot{s}_0$ will be constant as long as Δx_0, Δy_0, $\dot{\Delta x}$ and $\dot{\Delta y}$ are constant.

Because there are three unknowns, the underlying state equation must be three dimensional. The state equation is given by

$$\begin{bmatrix} s \\ \dot{s} \\ \ddot{s} \end{bmatrix}_{n+1} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ \dot{s} \\ \ddot{s} \end{bmatrix}_n . \qquad (43)$$

The observation equation can be obtained simply by squaring the range measurement, i.e.,

$$s^m_{n+1} = (\Delta r^m_{n+1})^2 . \qquad (44)$$

Equations (43) and (44) are in the form suitable for developing the so-called $\alpha\beta\gamma$ tracker algorithm. The algorithm is given below.

prediction:
$$\begin{bmatrix} s^+ \\ \dot{s}^+ \\ \ddot{s}^+ \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{s} \\ \dot{\hat{s}} \\ \ddot{\hat{s}} \end{bmatrix}_n ,$$

measurement error:
$$\tilde{s} = (\Delta r^m_{n+1})^2 - s^+ , \qquad (45)$$

state update:
$$\begin{bmatrix} \hat{s} \\ \dot{\hat{s}} \\ \ddot{\hat{s}} \end{bmatrix}_{n+1} = \begin{bmatrix} s^+ \\ \dot{s}^+ \\ \ddot{s}^+ \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta/\Delta t \\ \gamma/\Delta t^2 \end{bmatrix} \tilde{s}$$

The states are initialized using the first three consecutive measurements as follows

$$\begin{aligned} \hat{s}_2 &= s_2^m \quad , \\ \dot{\hat{s}}_2 &= [3\, s_2^m - 4\, s_1^m + s_0^m]/2\Delta t \quad , \\ \ddot{\hat{s}}_2 &= [s_2^m - 2\, s_1^m + s_0^m]/\Delta t^2 \quad . \end{aligned} \tag{46}$$

The gains are time scheduled according to

$$\alpha_n = \begin{cases} \dfrac{3\,(3n^2 + 3n + 2)}{D_n} & , \quad \text{for } n = 3 \text{ to } 15, \\ \alpha_{15} & , \quad \text{for } n > 15. \end{cases}$$

$$\beta_n = \begin{cases} \dfrac{18(2n+1)}{D_n} & , \quad \text{for } n = 3 \text{ to } 15, \\ \beta_{15} & , \quad \text{for } n > 15. \end{cases} \tag{47}$$

$$\gamma_n = \begin{cases} \dfrac{60}{D_n} & , \quad \text{for } n = 3 \text{ to } 15, \\ \gamma_{15} & , \quad \text{for } n > 15. \end{cases}$$

where

$$D_n = (n+1)(n+2)(n+3) \quad .$$

Figure 14 shows the time schedule characteristics for the three gains. All three gains begin with large values and decrease to smaller values as more measurements are incorporated in the estimates. The gains are prevented from becoming too small so as not to lose the filter "adaptability". These are characteristics exhibited by a Kalman filter.

As usual, when a measurement is missed because of surveillance failure, the predicted position, s^+, is used in place of the measurement (without advancing the gain computation time frame).

The range and range rate estimates are derived from $\hat{s}$ and $\dot{\hat{s}}$ as

$$\hat{\Delta r} = (\hat{s})^{1/2} \quad \text{and} \quad \dot{\hat{\Delta r}} = \dot{\hat{s}}/2\hat{\Delta r} \tag{48}$$

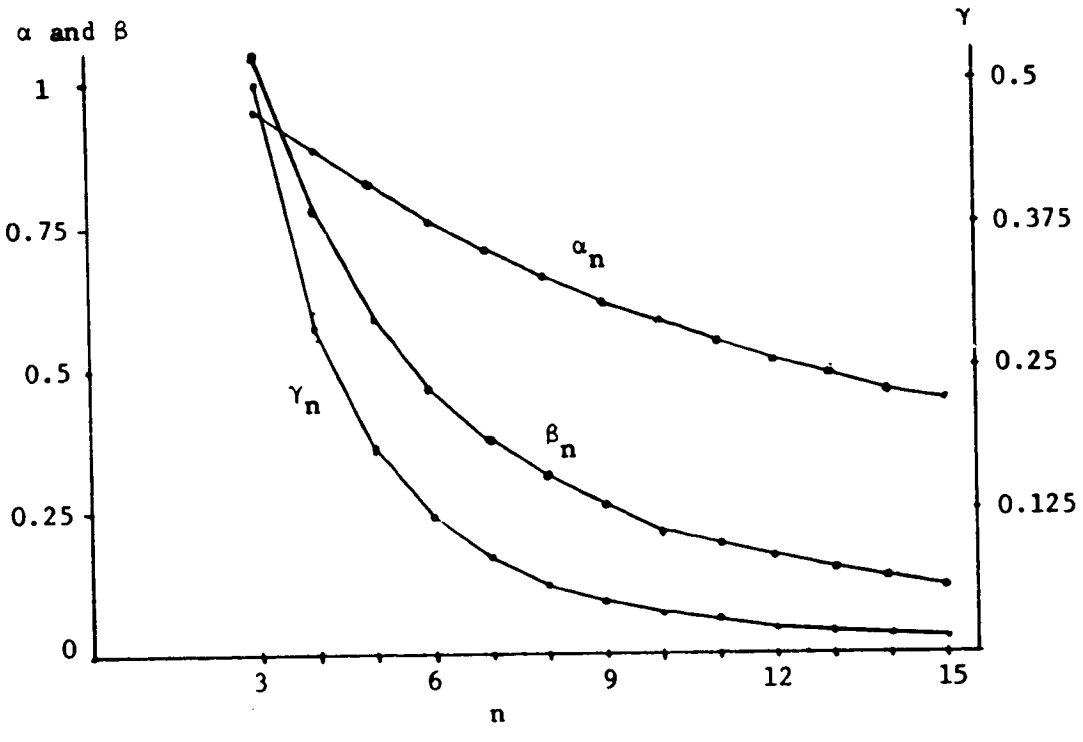


Figure 14. Gain Schedules for the Range Square αβγ Tracker Algorithm.

The range square tracker was designed (and successfully implemented in an experimental TCAS unit) to track targets equipped with transponders that had Mode A capability only. This does not preclude its usage for other targets, however.

The αβγ tracker design is a very ingenious realization. One comment needs to be added. In the pseudo-measurement Eq (44), the squared range measurement is defined by

$$s^m = (\Delta r^m)^2 \triangleq (\Delta r + \xi_r)^2 ,$$

where Δr is the true range and ξ_r is presumably an independent random number. Then, the above becomes

$$s^m = \Delta r^2 + 2\Delta r \xi_r + \xi_r^2 = s + \xi_s \ .$$

That is, the measured value is the sum of true value and "measurement noise", which sort of satisfies the usual filtering assumption. However, the noise, ξ_s, is no longer independent. It depends on the state, s. Therefore, the proposed $\alpha\beta\gamma$ tracker algorithm may not be optimal.

Correlated Error $\alpha\beta$ Tracker Up to now, we have assumed that the range measurement error is white noise, i.e., independent from one sample to the next. However, there is strong evidence that the error is strongly correlated [10]. If the low frequency (or bias) error term is removed, the error is governed by the following linear equation

$$\xi_{r,n+1} = \rho \, \xi_{r,n} + \eta_{n+1} \ . \tag{49}$$

Here,

ρ = correlation parameter with the empirical value of 0.6801, and
η = a zero mean stationary Gaussian noise with one sigma value of 69.5 ft.

The steady state variance of the range error is given by

$$\alpha_\xi^2 = (1-\rho^2)^{-1} \sigma_\eta^2 = (94.5 \text{ ft})^2 \ . \tag{50}$$

When the measurement error is white noise, it is true that the $\alpha\beta$ tracker given by Eq (40) performs near optimum when the variation in the range rate is negligible. However, the above is no longer valid when the error is known to be correlated, as in our present problem. A method is available to modify the original algorithm in order to take advantage of the fact that the error is correlated. The method is based on the optimum filter derived by Tarn [11].

The modification involves one additional term in the feedback signal. The modified closed loop filter equation is given by

$$\begin{bmatrix} \hat{\Delta r} \\ \hat{\dot{\Delta r}} \end{bmatrix}_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\Delta r} \\ \hat{\dot{\Delta r}} \end{bmatrix}_{n} + \begin{bmatrix} \alpha \\ \beta/\Delta t \end{bmatrix} \left\{ \Delta r^{m}_{n+1} - (\hat{\Delta r}_{n} + \Delta t \hat{\dot{\Delta r}}_{n}) - \rho(\Delta r^{m}_{n} - \hat{\Delta r}_{n}) \right\} \tag{51}$$

where ρ is the correlation parameter of the noise process.

It is noted that the modification requires one additional memory cell to store the previous range measurement.

It was shown that the above formulation does indeed result in performance improvement [12]. In a simulation study conducted earlier, the range rate estimation error was reduced by 23%. Furthermore, the formulation is fairly robust with respect to a small variation in ρ.

Monte Carlo Simulation Results

Simulation Scenarios Four of the previously discussed range filter algorithms were implemented in the Monte Carlo Simulation Program to obtain statistical performance data for the range and range estimation errors. The four algorithms are:

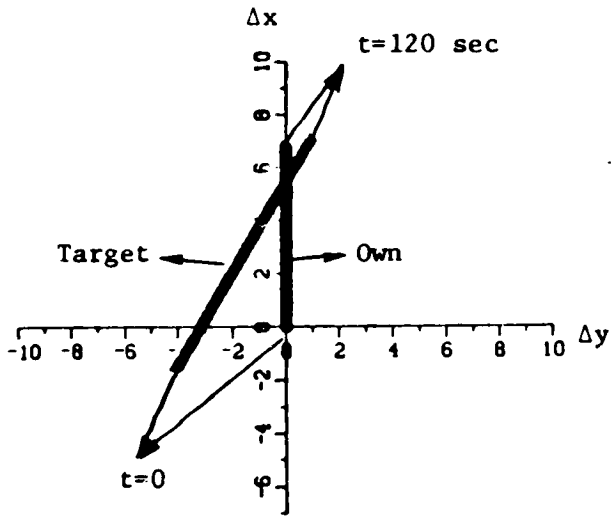
(1) Non-aided x-y fixed gain tracker;
(2) Target- and Own-aided Kalman filter;
(3) Range $\alpha\beta$ tracker; and
(4) Range square $\alpha\beta\gamma$ tracker.

Configuration (1) uses the proposed fixed gains (α = 0.67 and β = 0.25 for $\Delta\tau$ = 1 sec. Configuration (2) uses the Kalman gain update formula with the acceleration uncertainly of $\pm$ 16 fpss ($\pm 1\sigma$). Nominal measurement errors are assumed to be white noise processes with the standard deviation values of 75 ft and 1 deg for range and bearing respectively.

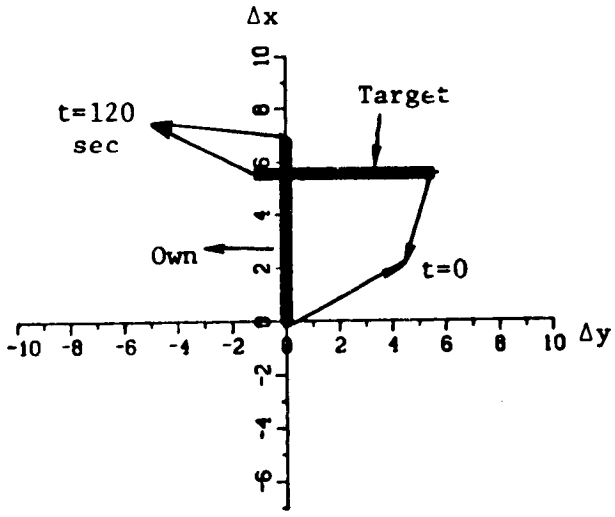
Four typical encounter scenarios were chosen motivated by the collision avoidance logic applications shown in Fig 15. These are (a) tail chase, (b) route crossing, (c) head-on, and (d) parallel turn-in encounters. Figure 15 shows the horizontal projections of each geometry in a north-east coordinate system. Own flies due north with a 200 kt ground speed starting from the origin. The target kinematic parameters - ground speed (V_G), heading (ψ) and the miss distance (M_d) - are listed in the following Table 6.

Statistical data were obtained based on sixty passes for each of the above encounter cases with nominal measurement errors for the nominal sampling period of 1 sec. To test the sensitivity of the estimates to measurement errors, the route crossing and parallel turn-in encounter cases were repeated with twice the nominal measurement errors; the head-on encounter was repeated with twice and four times the nominal values.

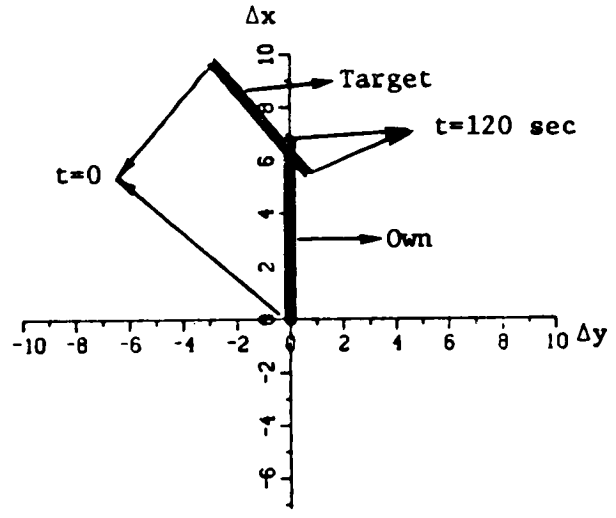
Except for the parallel turn-in case, these encounter scenarios are rather benign compared to the scenario used in Chapter II. However, these are



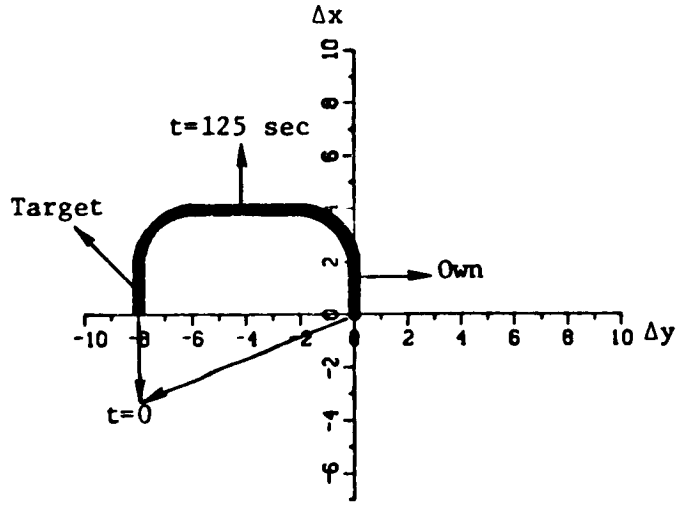
(a) Tail Chase



(b) Route Crossing



(c) Head-on



(d) Parallel Turn-in

Figure 15. Relative Kinematics for Test Cases

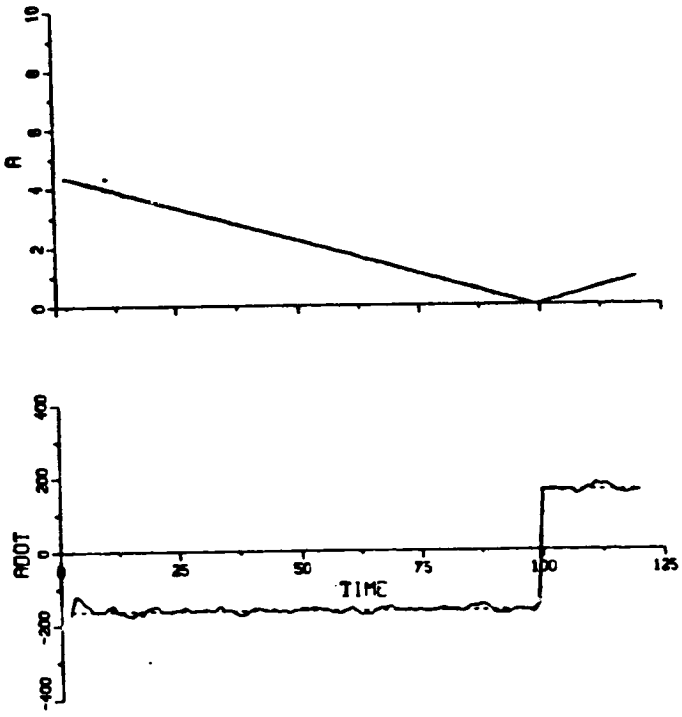
Table 6. Target Kinematic Parameters

Encounter	V_a (kt)	ψ (deg)	M_d (nmi)	Turn Rate (deg/sec)	Case No.	Figure No.
Tail Chase	300	30	0	0	a	15,16
Route Crossing	200	270	0	0	b.1 b.2	17,18 19,20
Head-on	160	140	0.3	0	c.1 c.2 c.3	21,22 23,24 25,26
Parallel Turn-in	200	0	0	1.5(target) -1.5(Own)	d.1 d.2	27,28 29,30

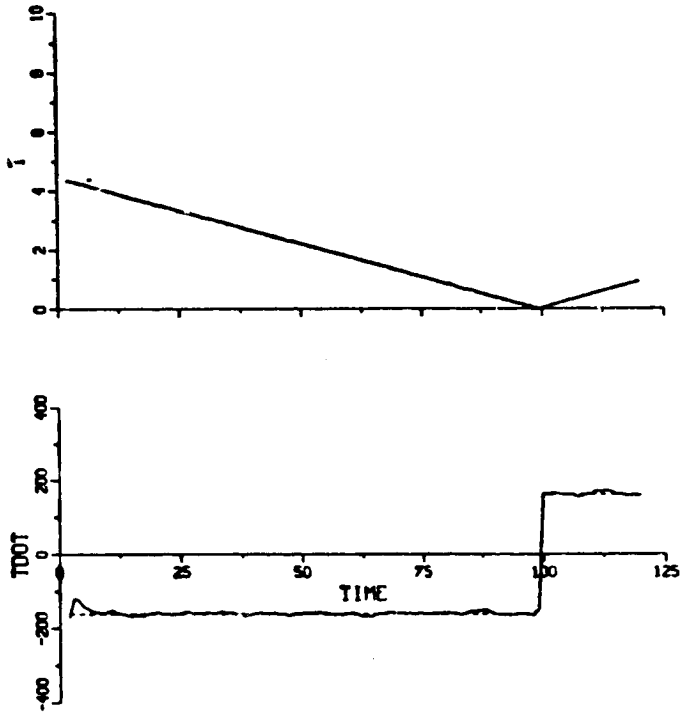
thought to represent typical CAS encounters. The proposed parallel turn-in scenario in the TCAS MOPS seems to be very severe. The scenario requires both Own and Target to perform 65 deg bank angle maneuvering at 600 kt ground speed resulting in more than 2g lateral accelerations. Therefore, this scenario was scaled down to reflect more representative values of commercial operations.

Simulation Results: Tail Chase Encounter Figure 16 shows sample time plots of range and range rate estimates together with the true values. Figure 17 shows the statistical time plots of error mean and mean $\pm 1\sigma$. Thus 68% of the error would fall between the dotted curves.

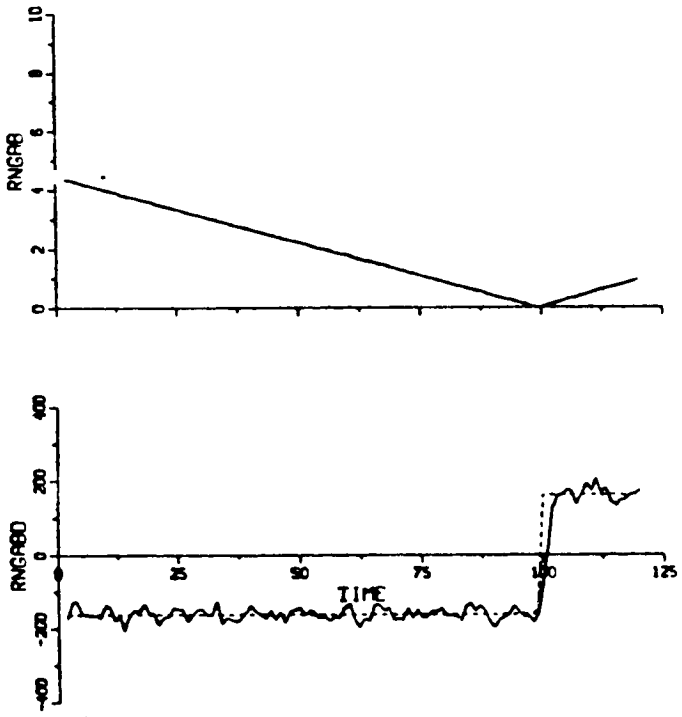
Range and range rate estimates derived from the x and y components of both non-aided and aided filters show very similar characteristics. The range errors are 44 and 41 ft and the rate errors are 6.8 and 6.4 kt respectively. The results follow the fact that the filter gains are comparable for both filters. The range rate error pulses appearing at approximately 100 sec are caused by division by very small range estimates.



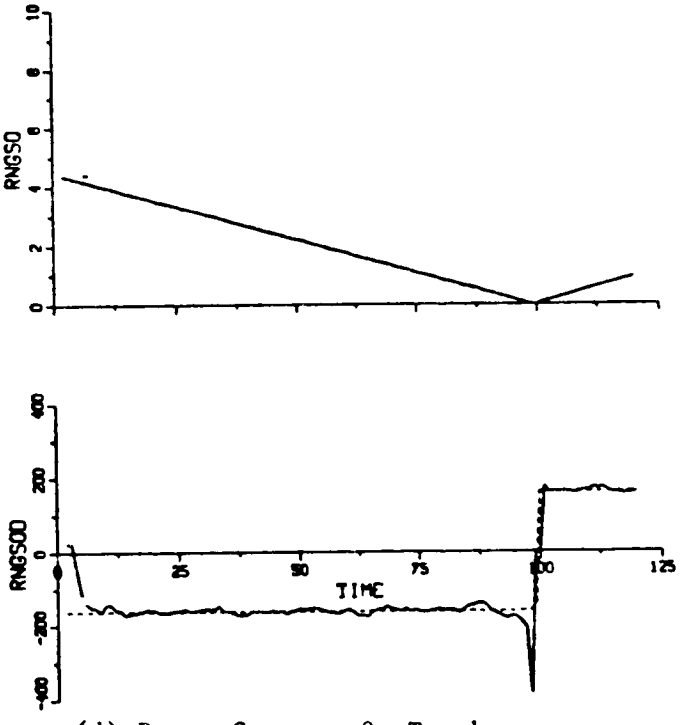
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter

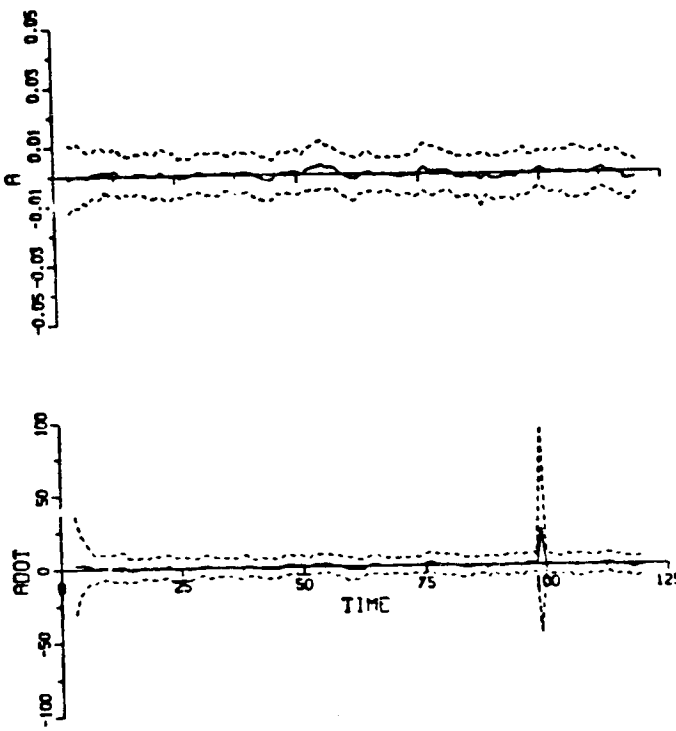


(c) Range αβ Tracker

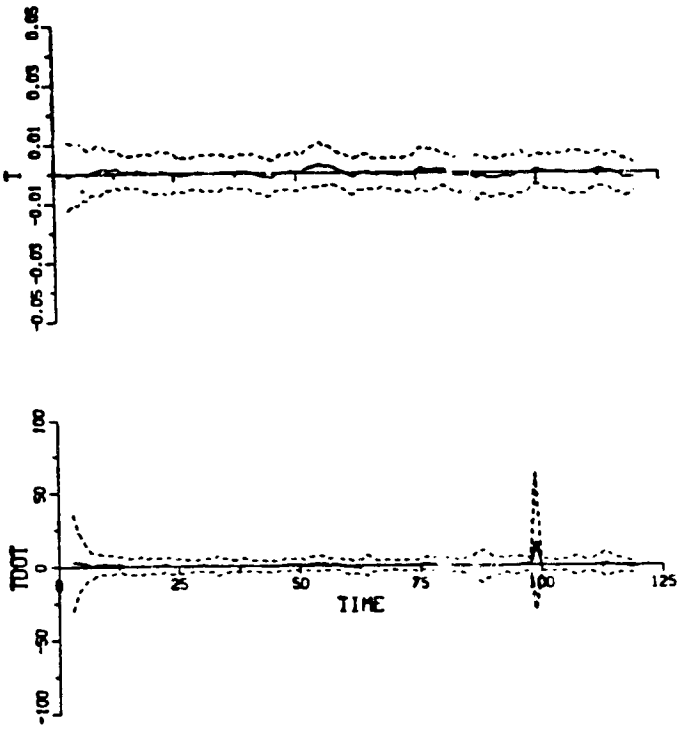


(d) Range Square αβγ Tracker

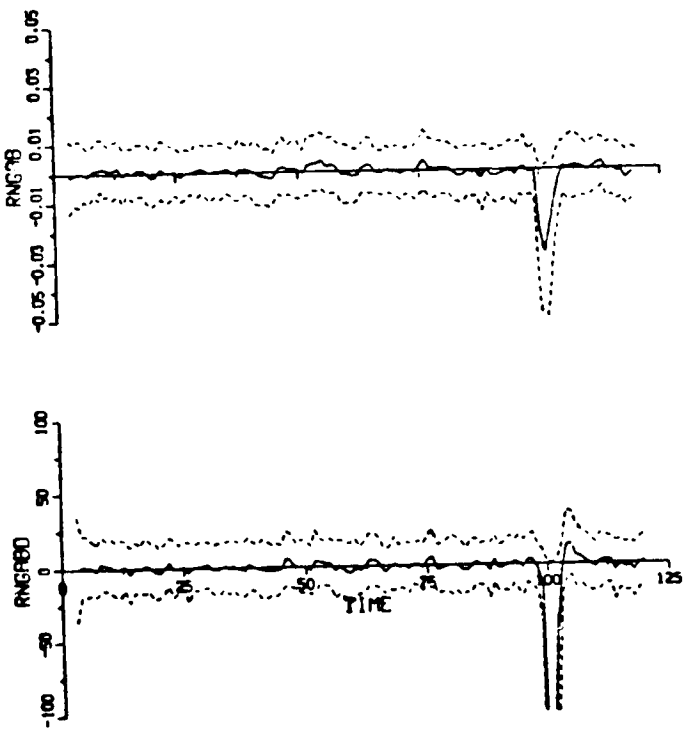
Figure 16. Sample Time Plots of Range and Range Rate Estimates for the Tail Chase Encounter.



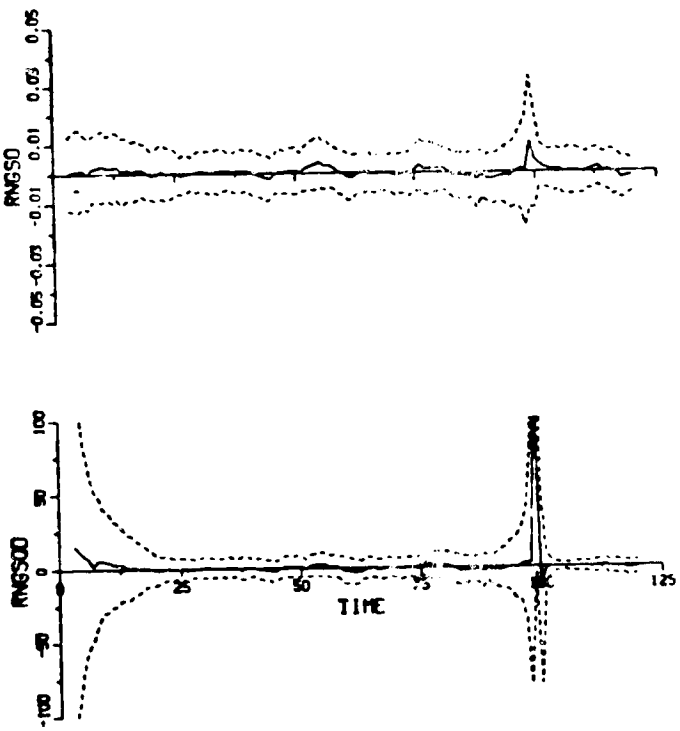
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter



(c) Range αβ Tracker



(d) Range Square αβ Tracker

Figure 17. Statistical Time Plots of Range and Range Rate for the Tail Chase Encounter.

The range αβ tracker shows somewhat larger errors -57 ft and 18.2 kt for the range and range rate. The rate error "pulse" at t=100 sec is not caused by division by small numbers. It represents dynamic delay error due to the sign change in the range rate at that particular time.

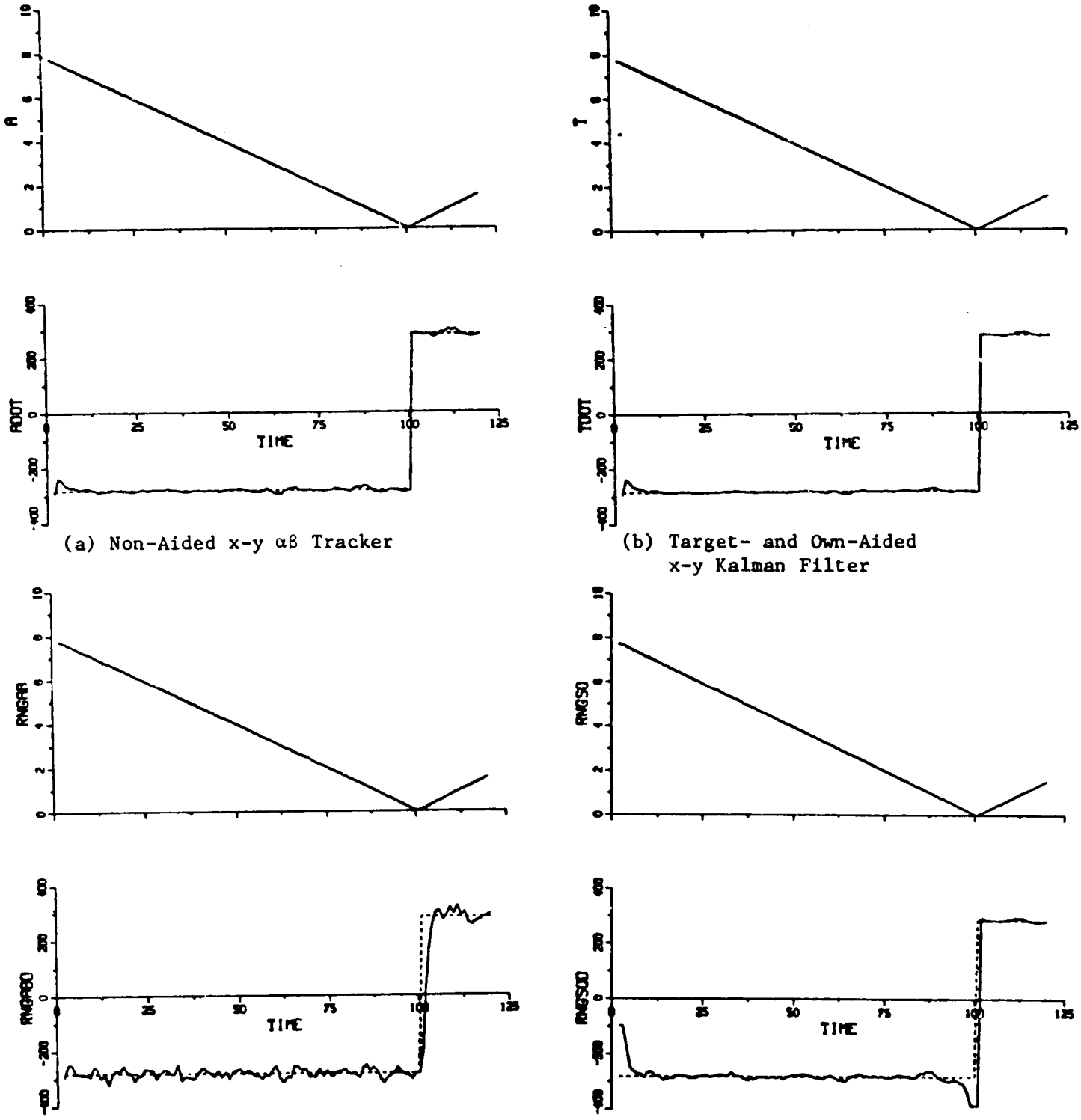
The range square αβγ tracker shows the steady state errors of 80 ft and 9.0 kt. When the range is zero, the range rate error shows a peculiar doublet behavior. The effect of the gain reduction schedule is very apparent in the initial transient of the rate estimate. The filter settles down to the steady state operation at $t \approx 20$ sec as expected.

Route Crossing Encounter Time plots of the simulation results, for the route crossing encounter, are shown in Figs 18 through 21. The previous comments generally apply to this case also. The error statistics are very similar to the previous case.

Figures 20 and 21 show the results with the measurement error magnitudes twice those of the nominal case. Compared to the nominal case, the error standard deviations are twice as large.

The non-aided and the aided configurations show similar mean error characteristics in range with a maximum of 60 ft. Since the other two configurations do not show a similar symptom, the mean error is caused by the y axis component. Because the target track is due west, whereas Own's is due north, the bearing error affects the y axis directly which induces the range effect.

The singularity problem (of dividing by small range) becomes worse for the range square tracker as the noise level becomes larger. This implies that the range square filter is not reliable at extremely small range. The range αβ tracker suffers the same reliability problem caused by an entirely different effect - the dynamic delay error build up. One other problem encountered for the range square αβγ tracker is the problem of the "range square" state becoming negative. This causes the computer implementation problem of taking the square-root of a negative number.



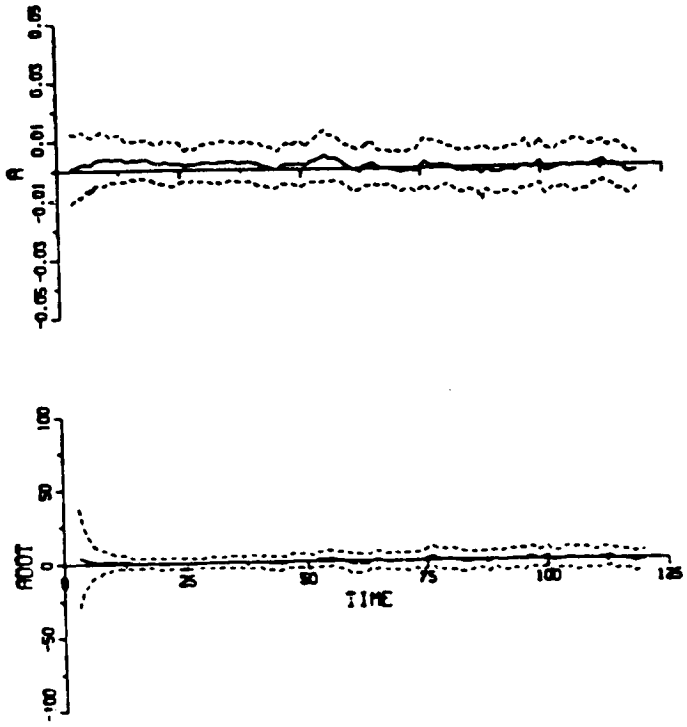
(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

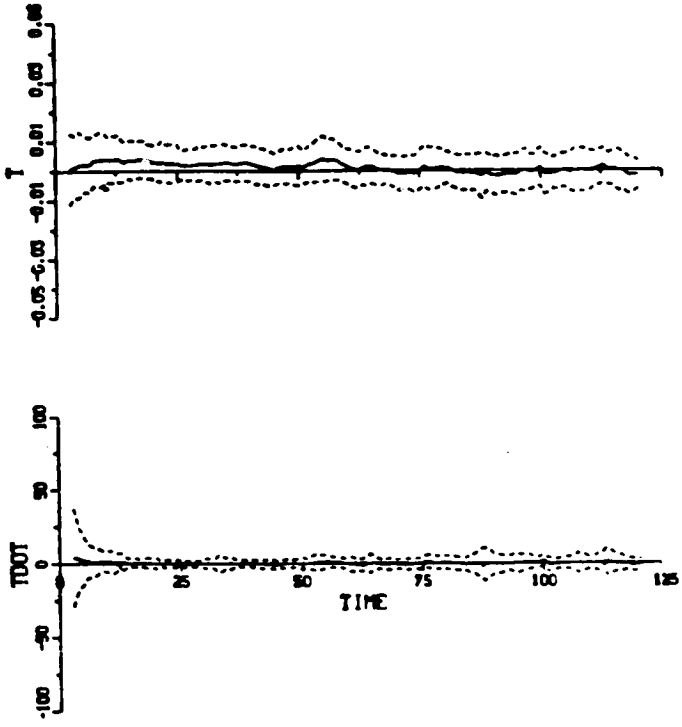
(c) Range αβ Tracker

(d) Range Square αβγ Tracker

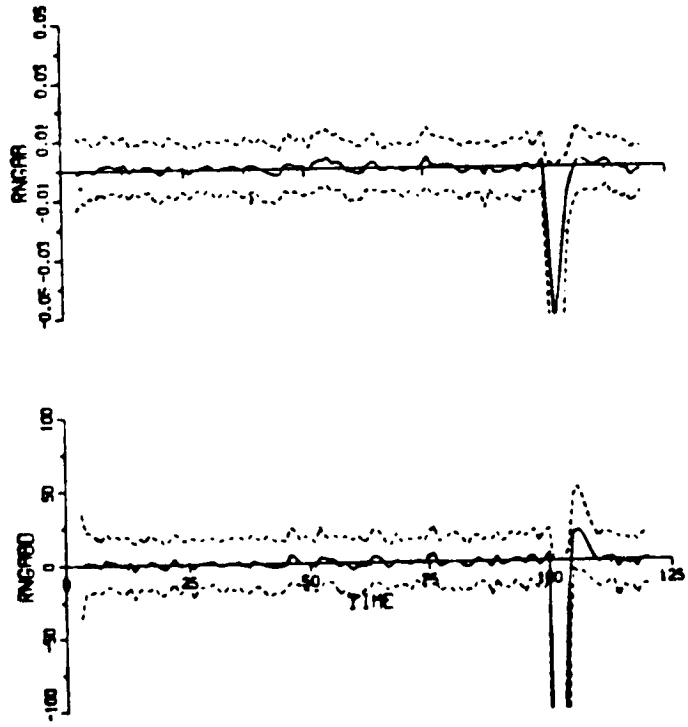
Figure 18. Sample Time Plots of Range and Range Rate Estimates for the Route Crossing Encounter.



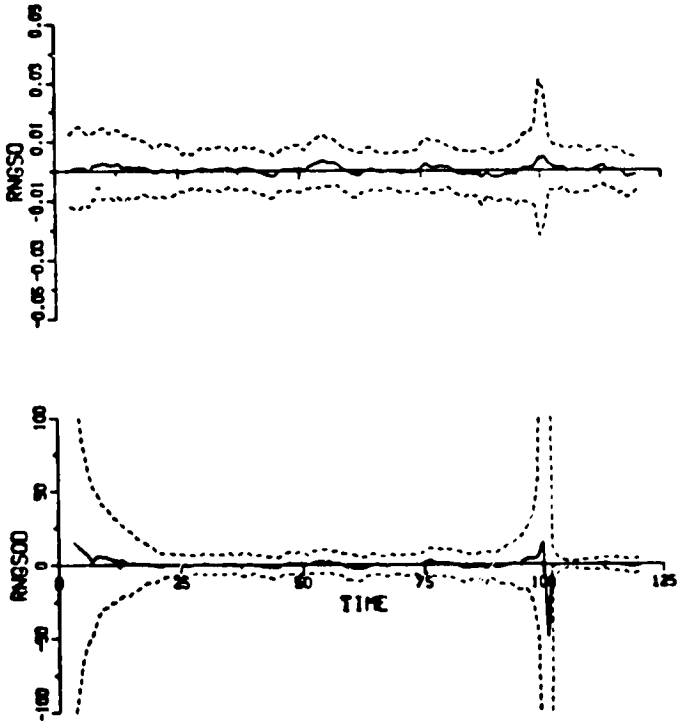
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter

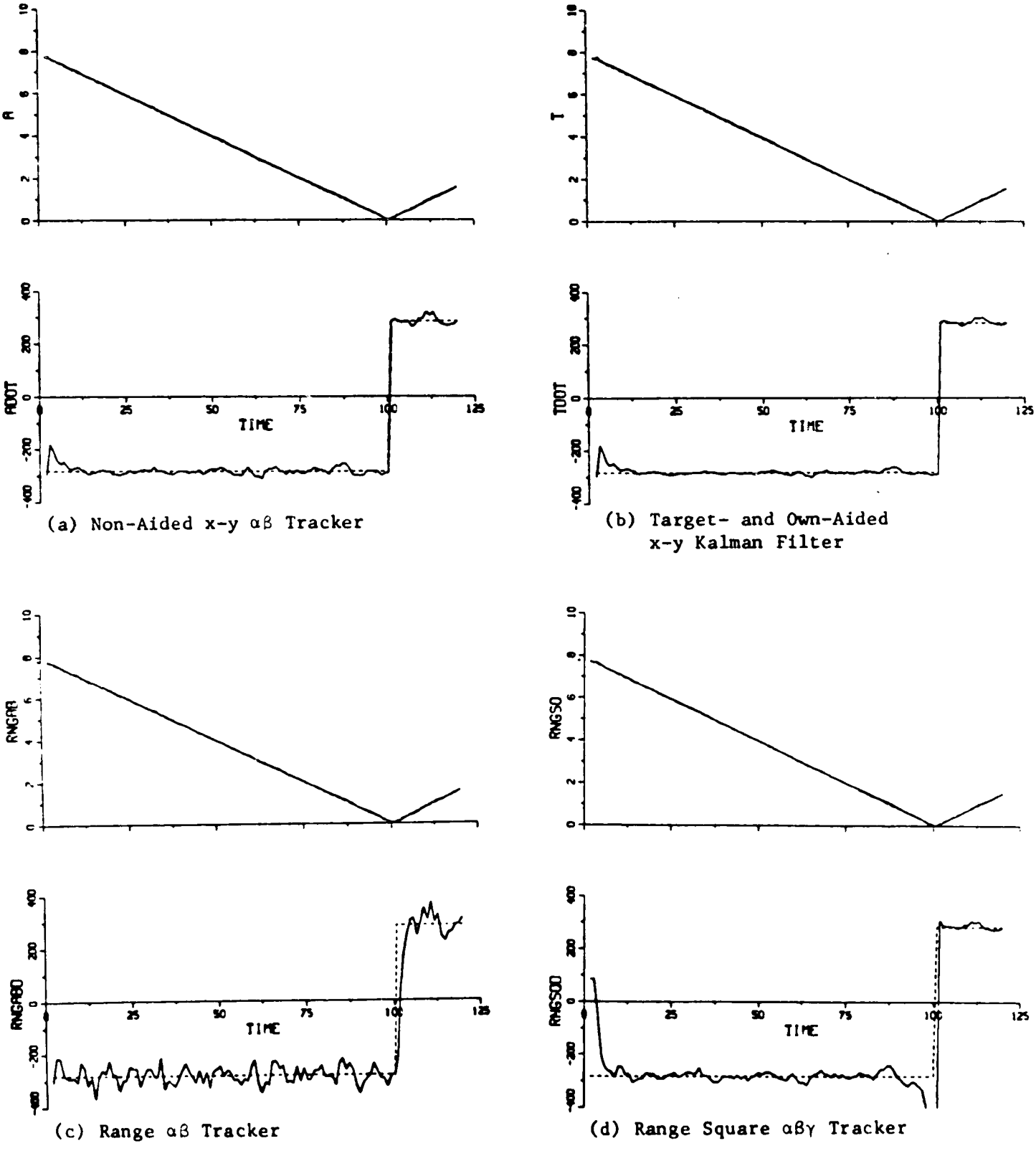


(c) Range αβ Tracker



(d) Range Square αβγ Tracker

Figure 19. Statistical Time Plots of Range and Range Rate Errors for the Route Crossing Encounter.



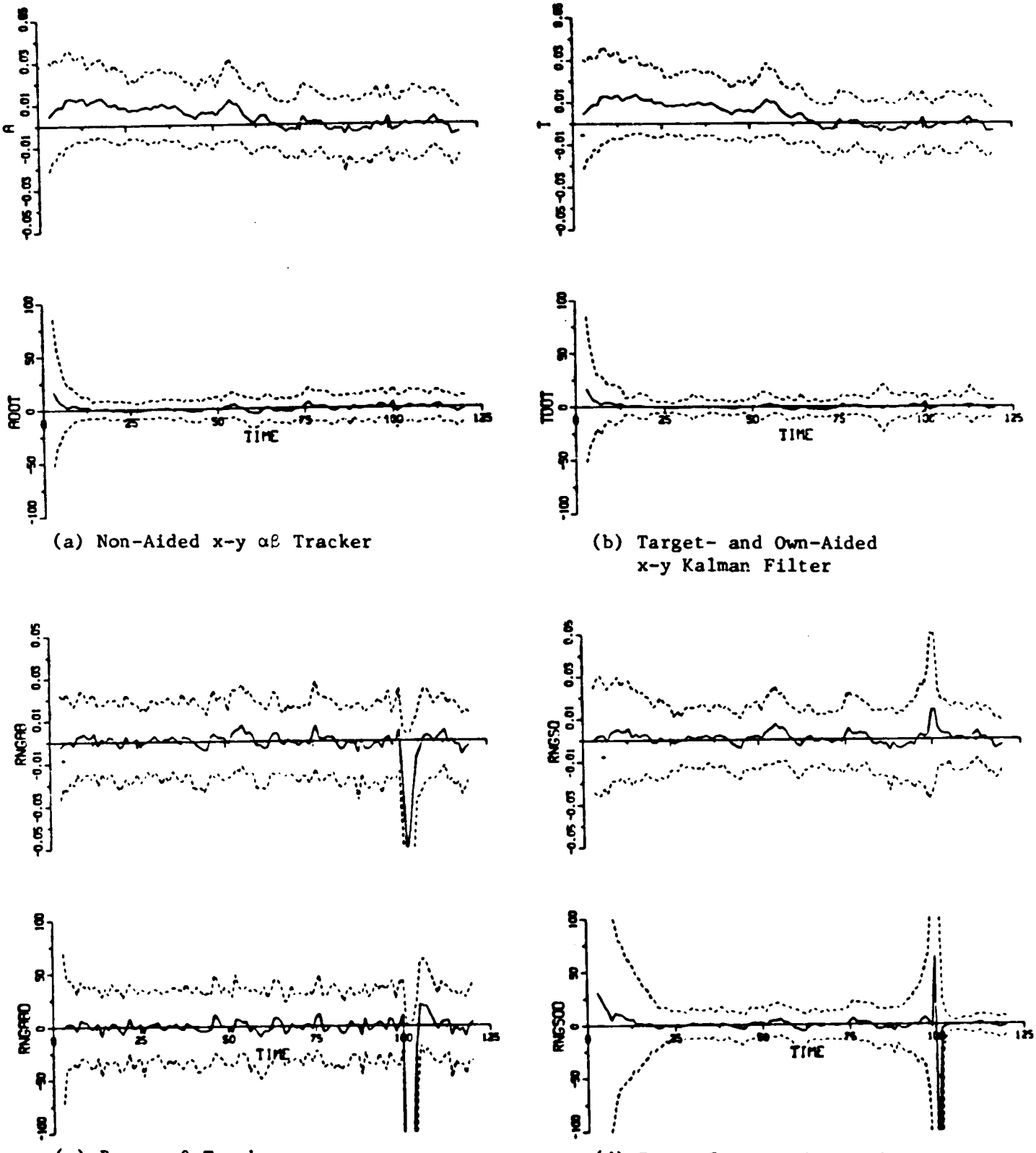
(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

(c) Range αβ Tracker

(d) Range Square αβγ Tracker

Figure 20. Sample Time Plots of Range and Range Rate Estimates for the Route Crossing Encounter (twice the nominal errors).



(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

(c) Range αβ Tracker

(d) Range Square αβγ Tracker

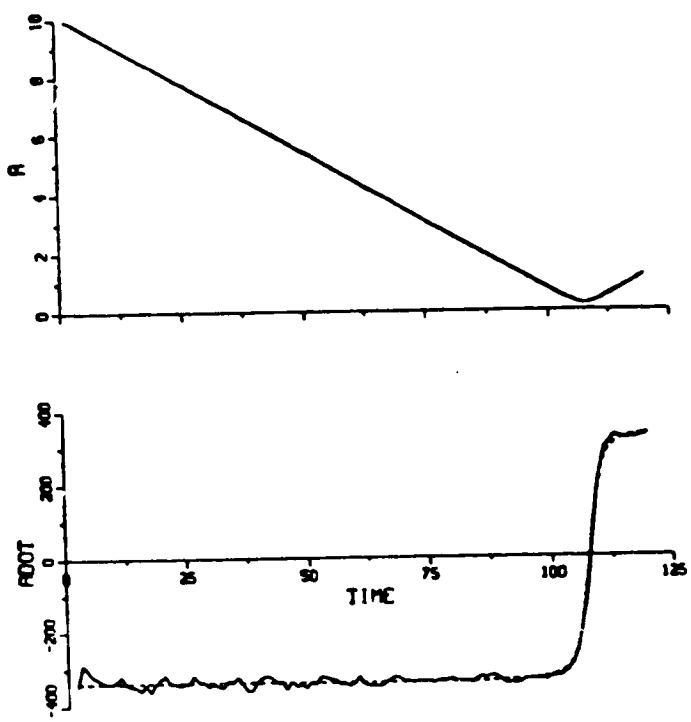
Figure 21. Statistical Time Plots of Range and Range Rate Errors for the Route Crossing Encounter (twice the nominal error).

Head-on Encounter Figures 22 through 27 show the results of the head-on encounter simulation results. The measurement errors were varied twice and four times the nominals. The largest level represents the lower performance enhanced TCAS II capability. As before, the previous comments apply here mostly. Because the encounter does not involve a collision (the miss distance is 0.3 nmi), the singularity problem does not manifest in the estimates. The $\alpha\beta$ tracker shows a considerable dynamic delay error when the range is closed to the minimum.

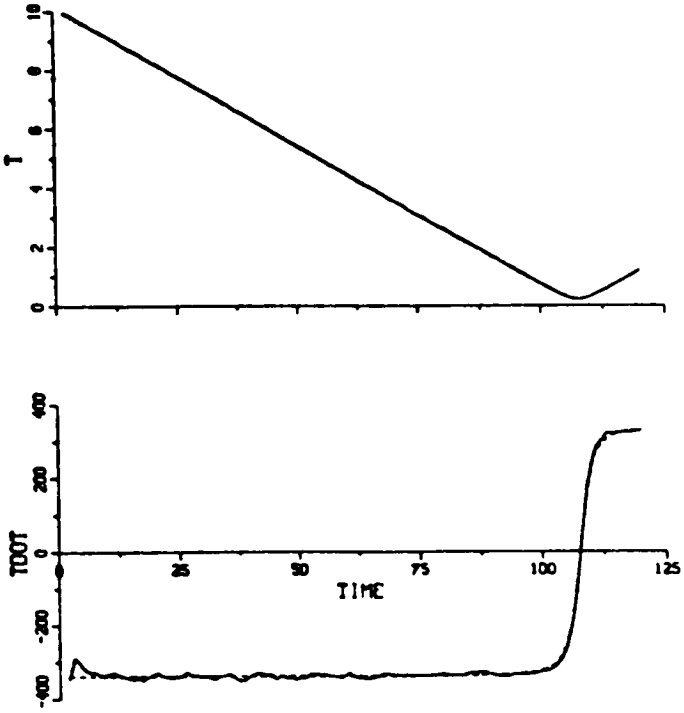
As can be expected, general performance of the estimation algorithms degrade as the measurement noise levels increase. In fact the standard deviations are remarkably proportional. The range errors varied from 44.4 - 173.4 ft for the aided configuration to 54.7 - 228.1 ft for the $\alpha\beta$ tracker. The rate errors vary from 6.8 - 28.5 kt for the aided configuration to 18.5 - 73.2 kt for the $\alpha\beta$ tracker. The former provided the most accurate estimates; the latter provides the poorest estimates.

The range effect is apparent in the x-y based estimates, but it is not in the other two configurations. This should be intuitively understood because the latter two do not contain the bearing measurement whereas the first two do.

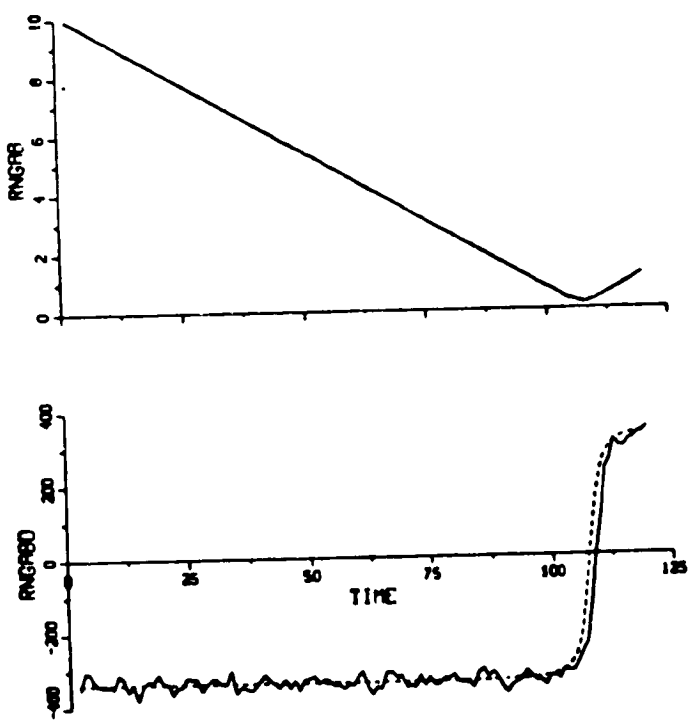
In these rectilinear trajectory cases, the estimates based on the non-aided and the Target- and Own-aided configurations as well as the range square $\alpha\beta\gamma$ tracker do not show substantial qualitative difference (except the singularity problem of the last tracker). The aided estimates show the smallest errors. Thus, if the underlying kinematics are rectilinear, then all three configurations would be equally effective. This follows from the fact that the filter model equations are exact; hence, no induced dynamic error.



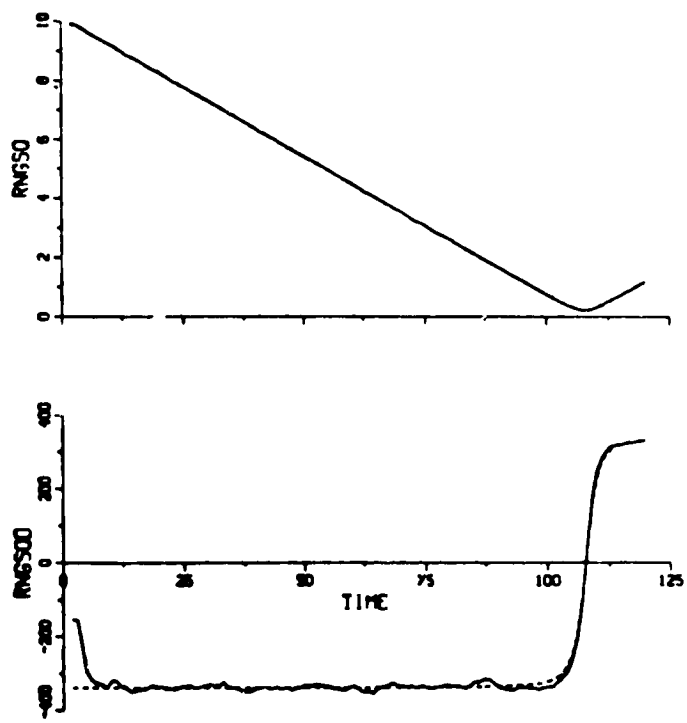
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter

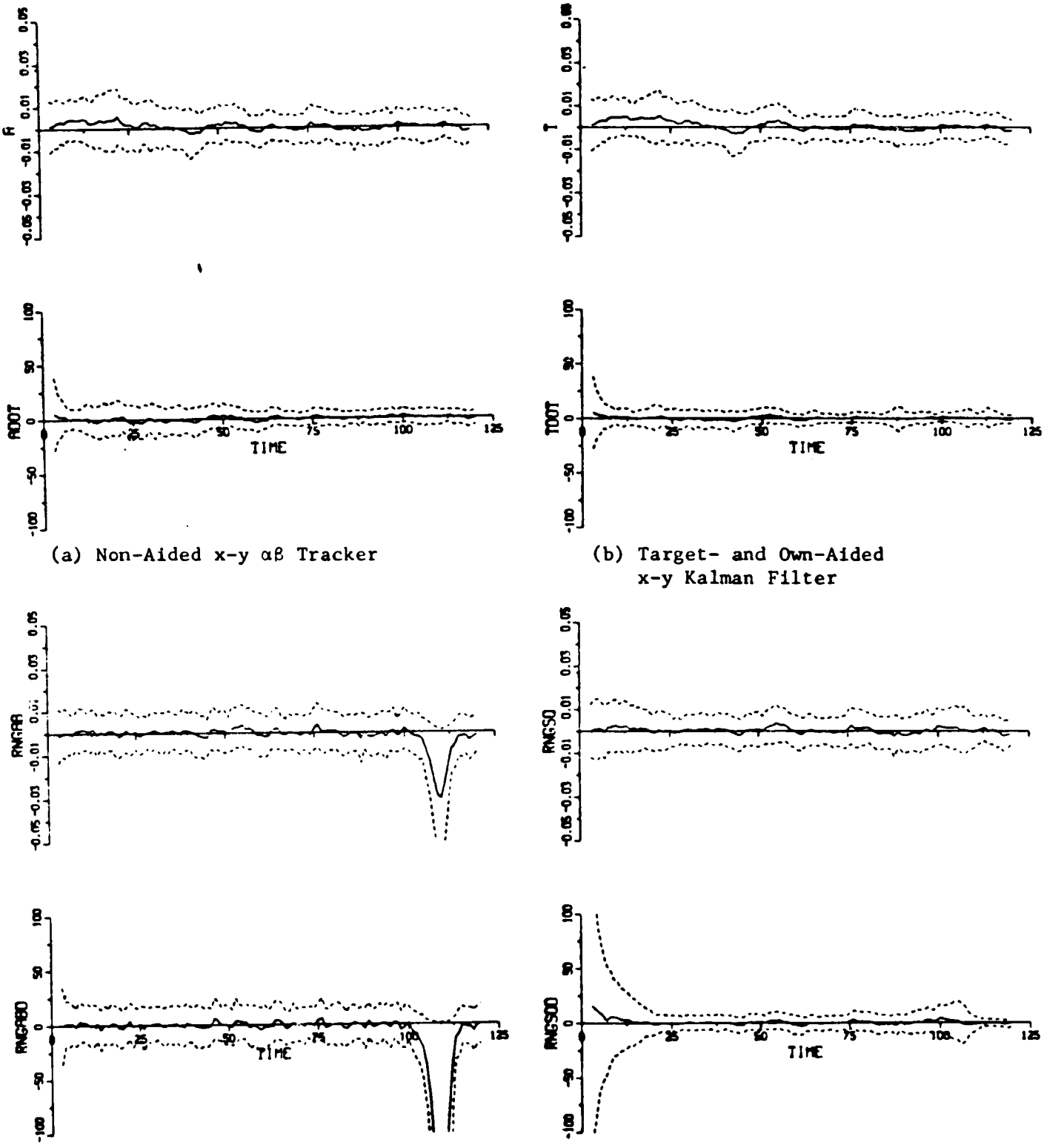


(c) Range αβ Tracker



(d) Range Square αβγ Tracker

Figure 22. Sample Time Plots of Range and Range Rate Estimates for the Head-on Encounter.



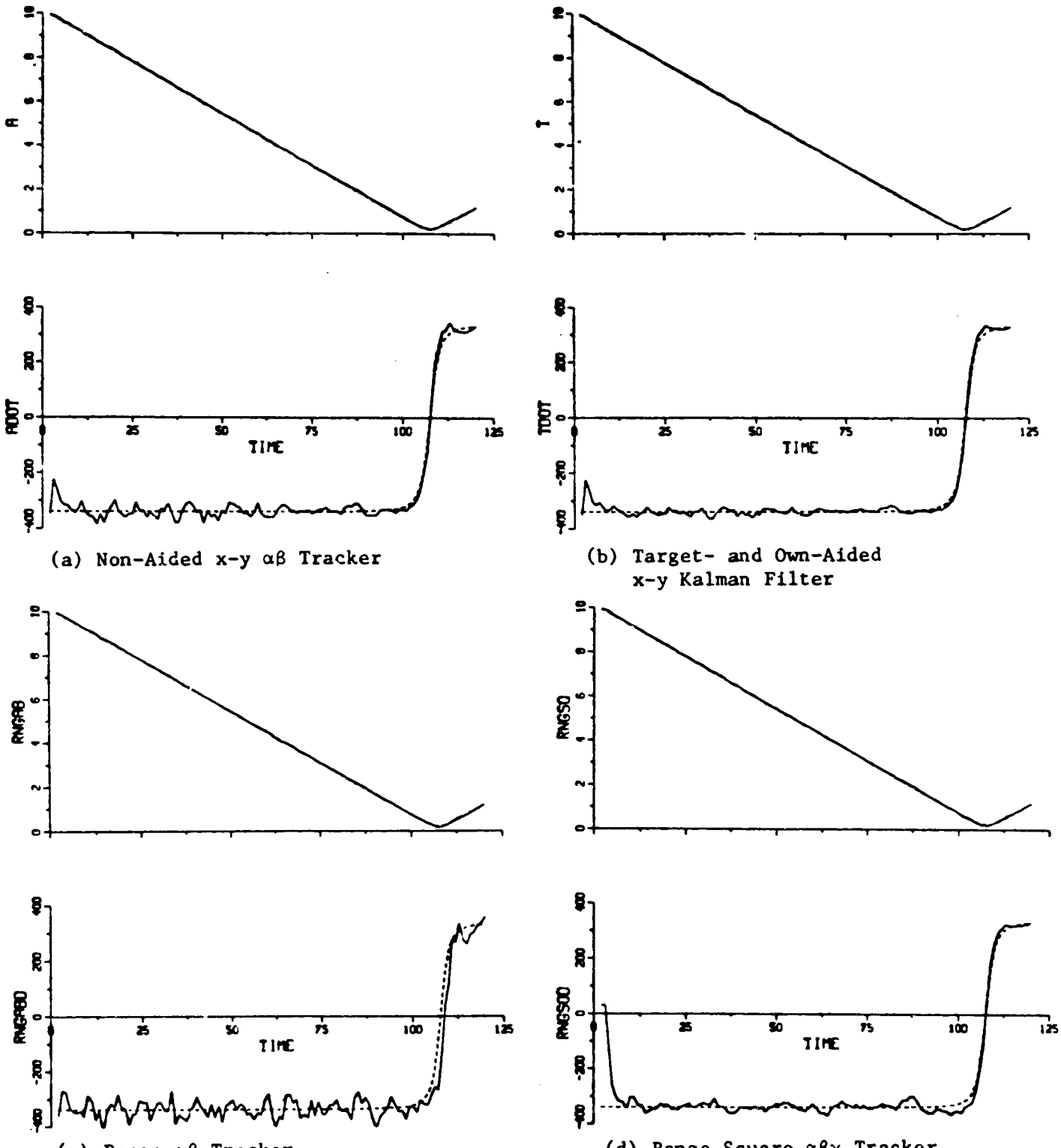
(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

(c) Range αβ Tracker

(d) Range Square αβγ Tracker

Figure 23. Statistical Time Plots of Range and Range Rate Errors for the Head-on Encounter.



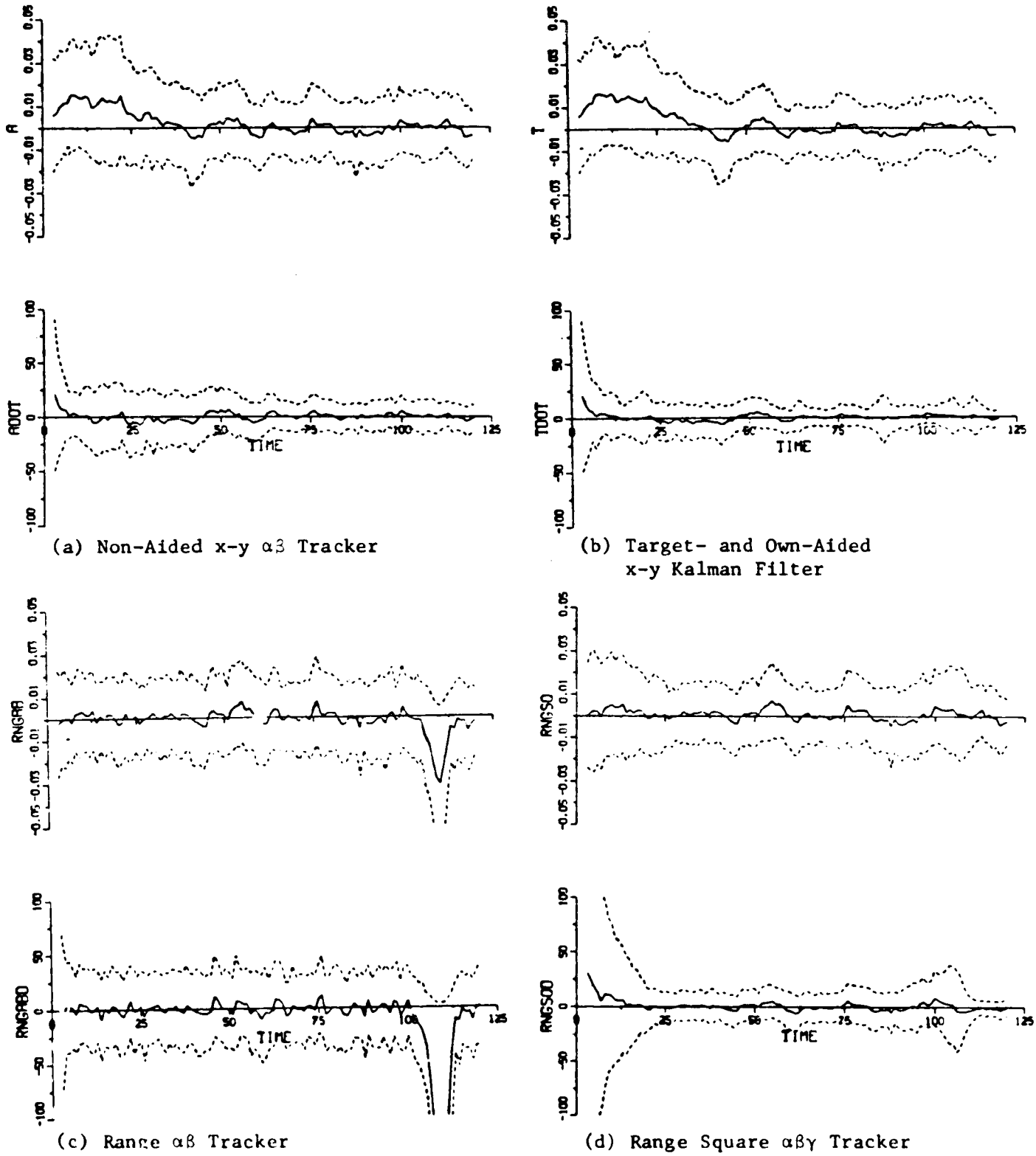
(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

(c) Range αβ Tracker

(d) Range Square αβγ Tracker

Figure 24. Sample Time Plots of Range and Range Rate Estimates for the Head-on Encounter (twice the nominal errors).



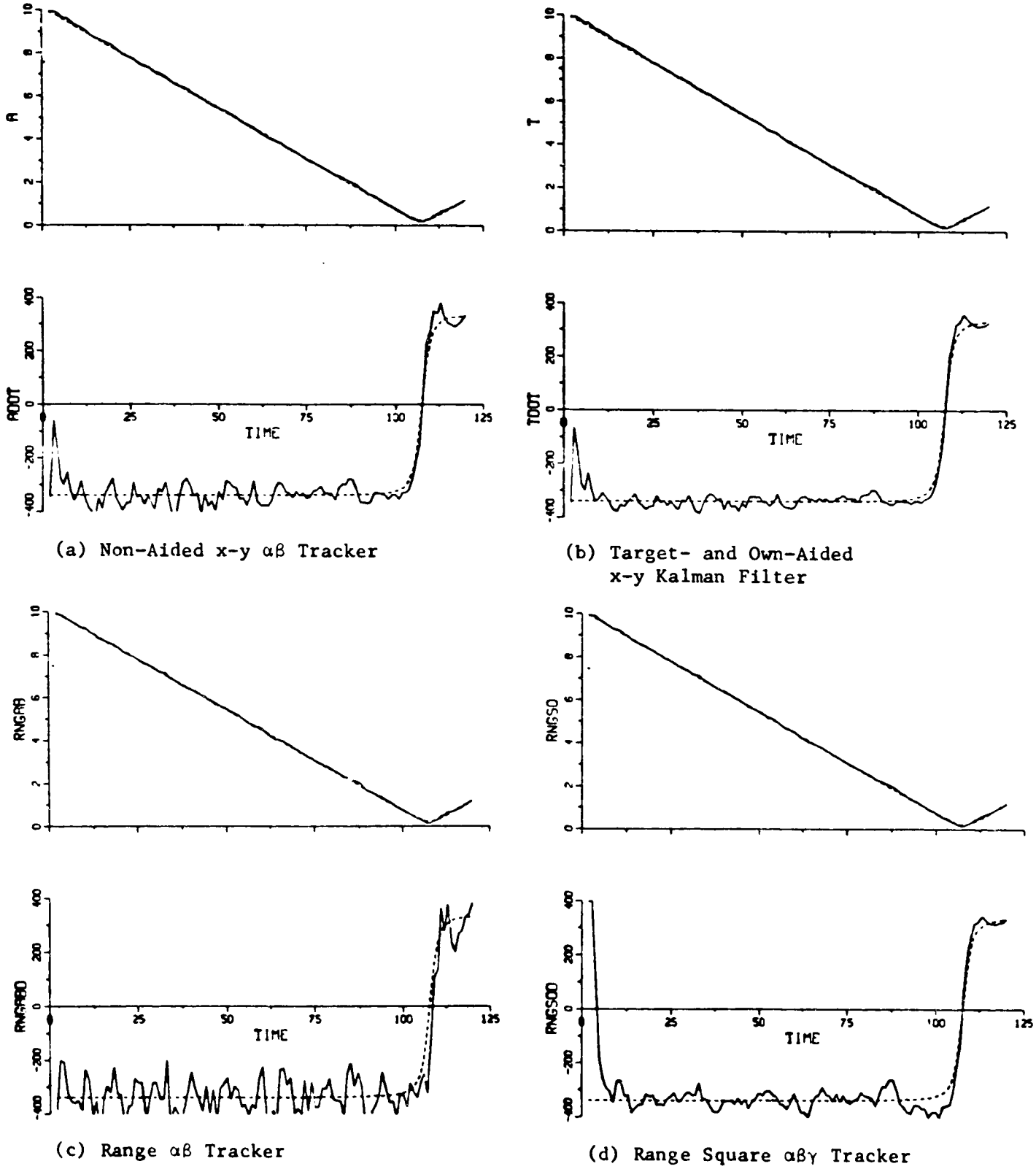
(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

(c) Range αβ Tracker

(d) Range Square αβγ Tracker

Figure 25. Statistical Time Plots of Range and Range Rate Errors for the Head-on Encounter (twice the nominal errors).



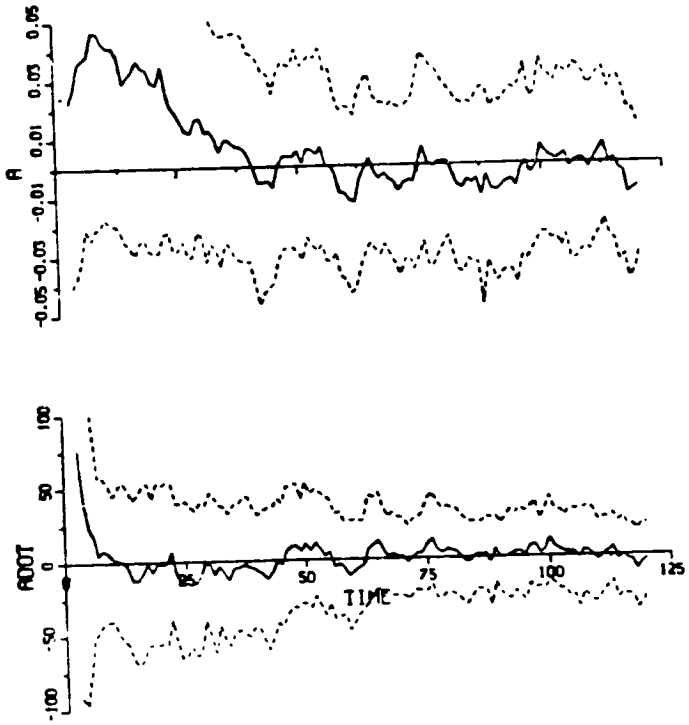
(a) Non-Aided x-y αβ Tracker

(b) Target- and Own-Aided x-y Kalman Filter

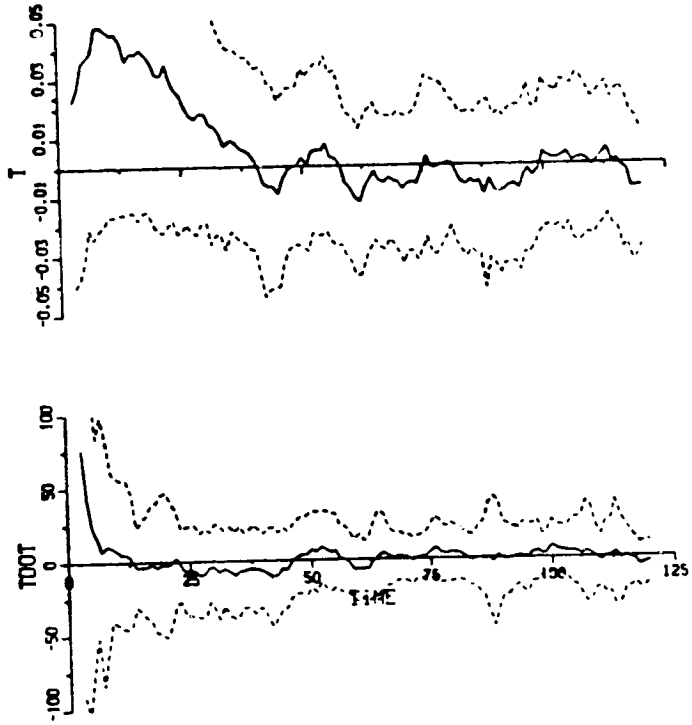
(c) Range αβ Tracker

(d) Range Square αβγ Tracker

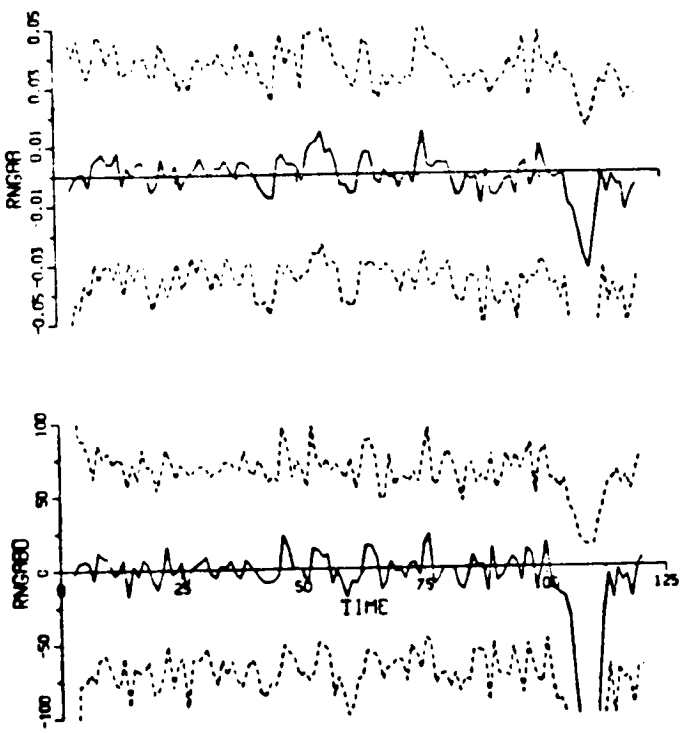
Figure 26. Sample Time Plots of Range and Range Rate Estimates for the Head-on Encounter (four times the nominal errors).



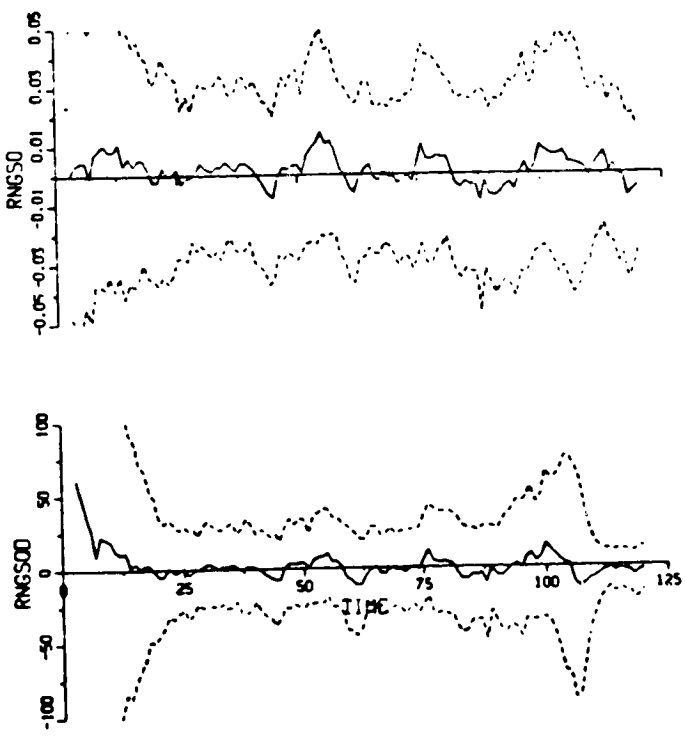
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter



(c) Range αβ Tracker



(d) Range Square αβγ Tracker

Figure 27. Statistical Time Plots of Range and Range Rate Errors for the Head-on Encounter (four times the nominal errors).

Parallel Turn-in Encounter Figures 28 through 31 show the simulation results for the parallel turn-in encounter. The encounter scenario involves both aircraft turning with 15 deg bank angle maneuvers. The combined acceleration is 0.54 g. Figure 28 shows that the range rate is no longer constant, and the range is not rectilinear.

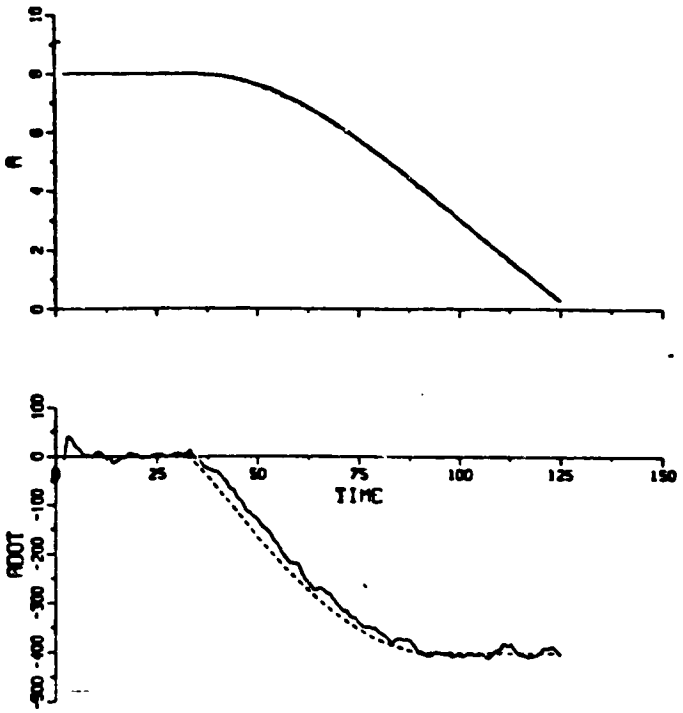
Now, by referring to Figs 29 and 31, the following comments can be made:

(i) The non-aided x-y tracker shows substantial mean errors with the maximum range error of 91 ft and the range rate error of 35.1 kt. The standard deviations are 93 ft and 15.0 kt for the nominal measurement error level. Surprisingly, the error magnitudes do not increase proportionally. This may be due to the fact that the number of the Monte Carlo passes are too small to provide the statistical precision. Note that the error curves contain more "roughness" for the higher error noise.

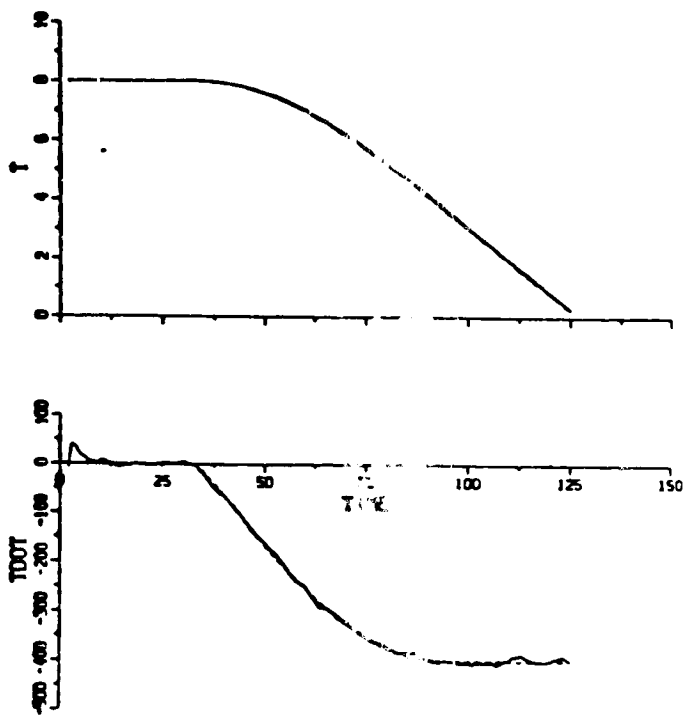
(ii) The aided configuration does not show outstanding mean errors. The standard deviations are comparable to other encounter cases.

(iii) The mean errors for the $\alpha\beta$ tracker are substantially smaller. The standard deviations are comparable to other encounter cases. This implies that the gain values are selected more on the basis of "tracking" rather than on the basis of "smoothing". During the maneuver, this tracker performs better than the other two non-aided configurations.

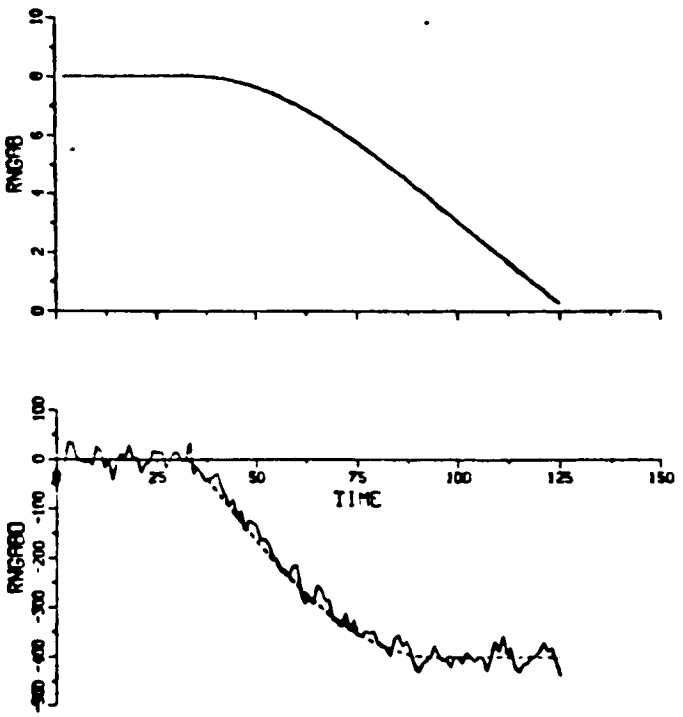
(iv) The mean errors of the range square tracker show interesting features. The maximum mean errors were 82 ft and 159 ft for the range and 29 kt for the range rate. The standard deviations do not show substantial increase. The first portion is caused by the dynamic delay not being able to catch up with the acceleration. The mean errors show oscillation. The second portion is caused by the effect of low gains, i.e., the tracker can not catch up because the low gains prevent rapid recovery.



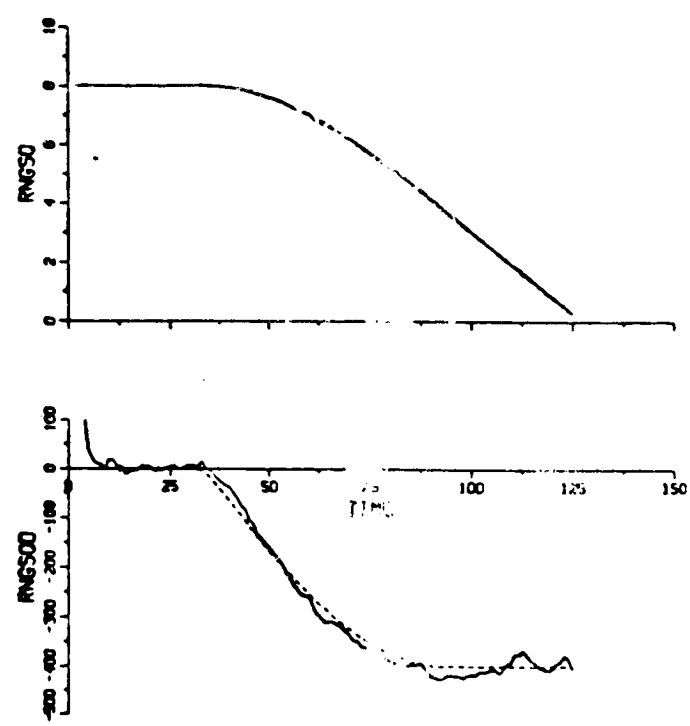
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter

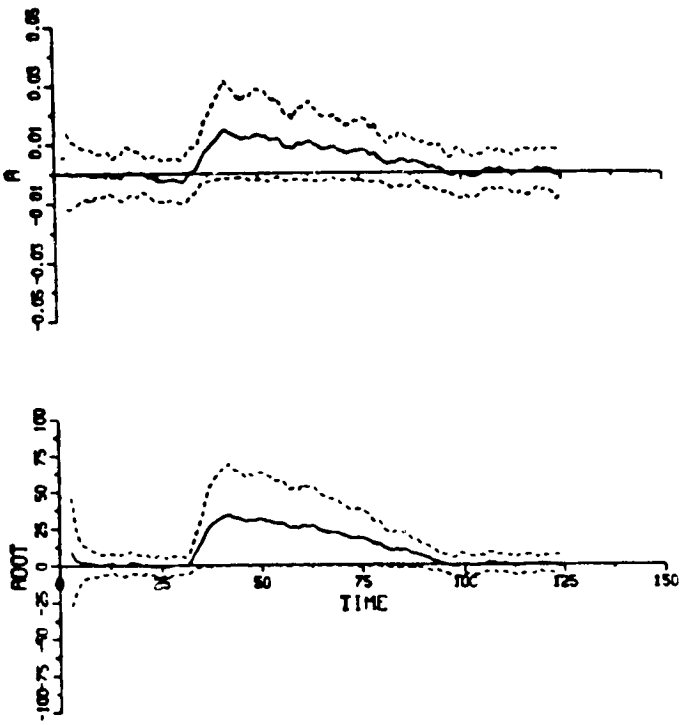


(c) Range αβ Tracker

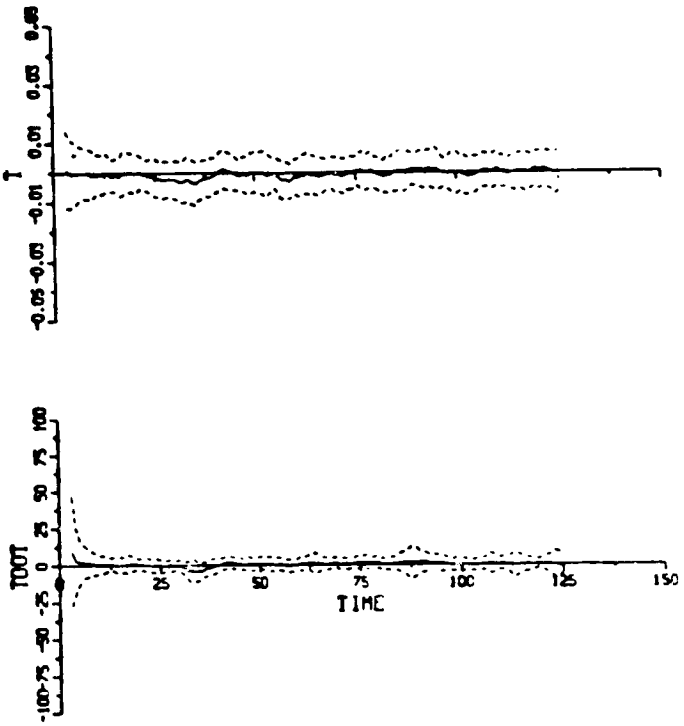


(d) Range Square [illegible] Tracker

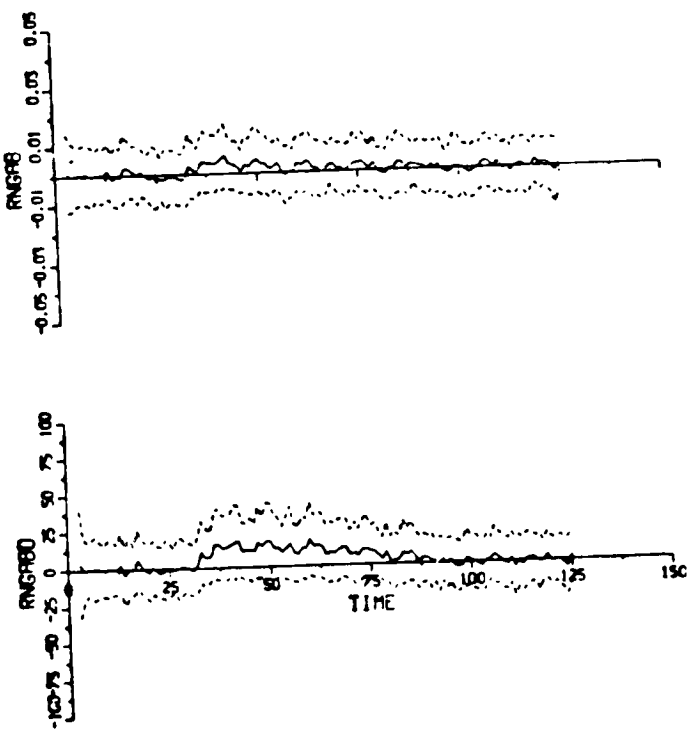
Figure 28. Sample Time Plots of Range and Range Rate Estimates for the Parallel Turn-in Encounter.



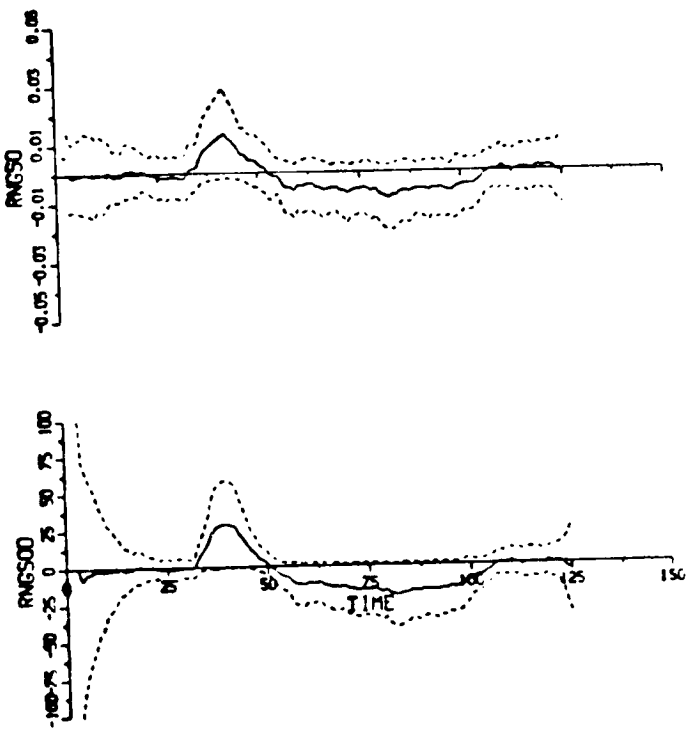
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter

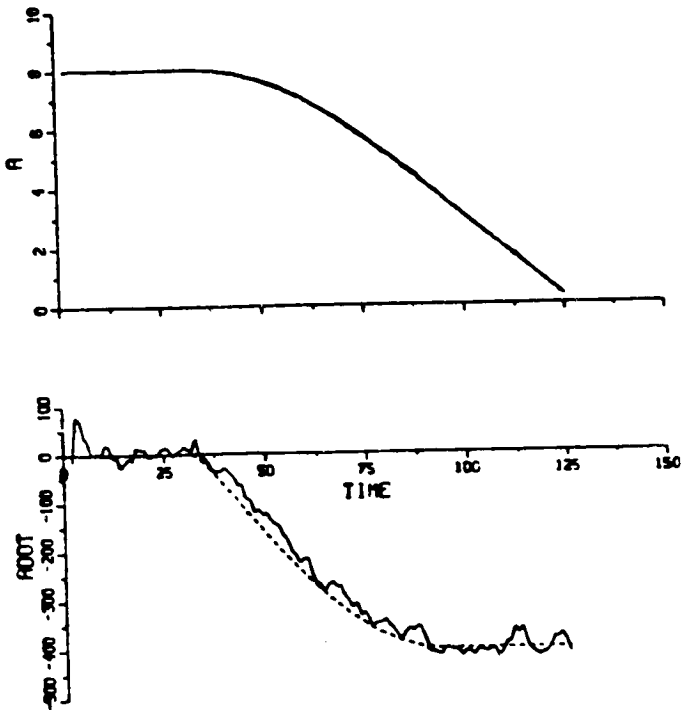


(c) Range αβ Tracker

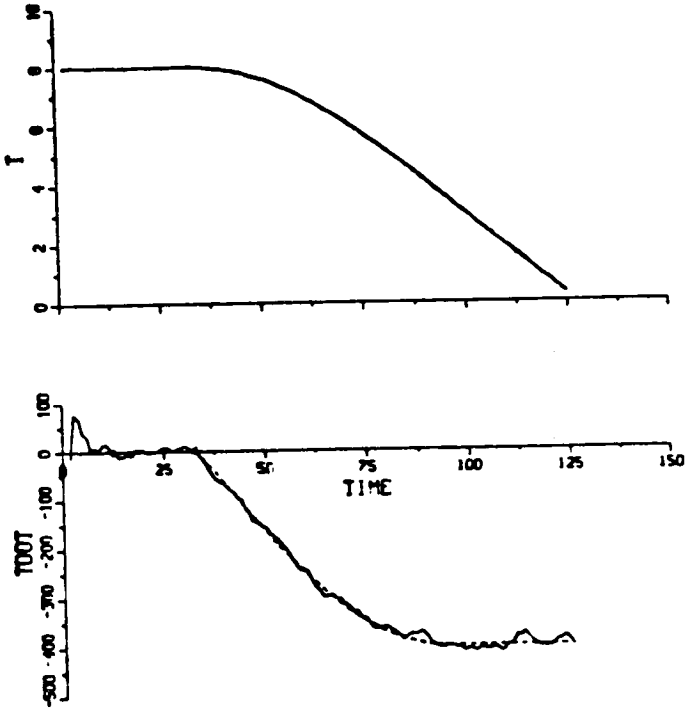


(d) Range Square αβγ Tracker

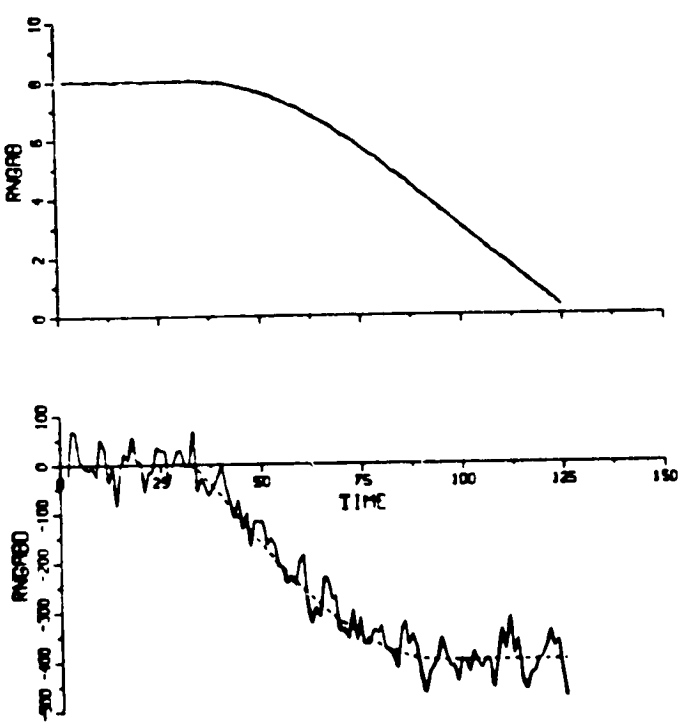
Figure 29. Statistical Time Plots of Range and Range Rate Errors for the Parallel Turn-in Encounter.



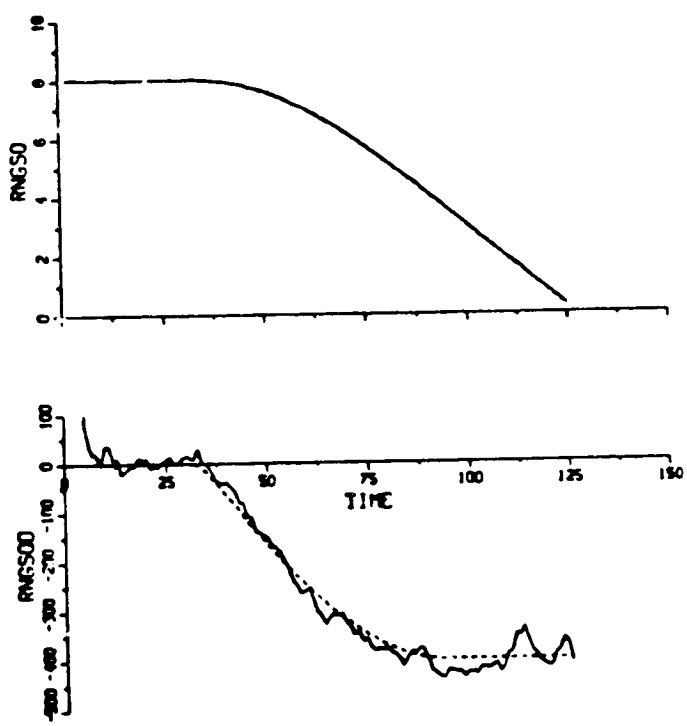
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter

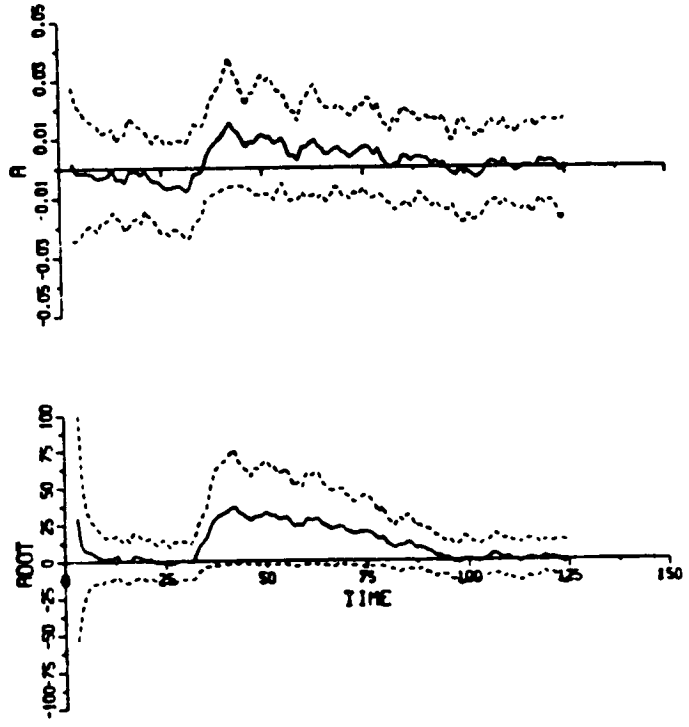


(c) Range αβ Tracker

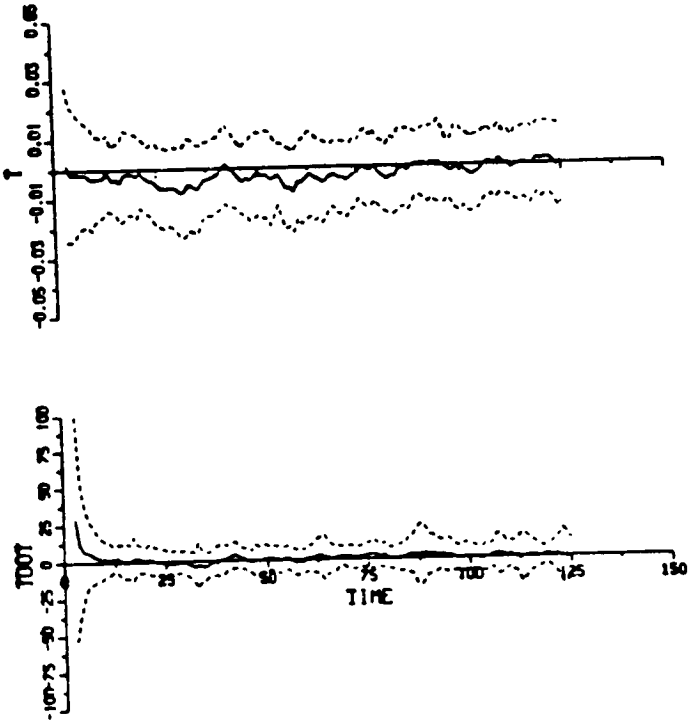


(d) Range Square αβγ Tracker

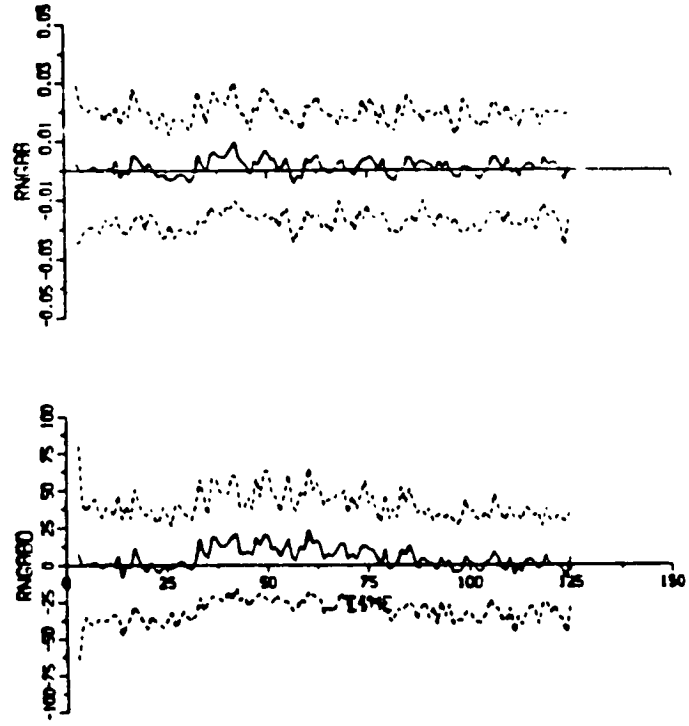
Figure 30. Sample Time Plots of Range and Range Rate Estimates for the Parallel Turn-in Encounter (twice Nominal).



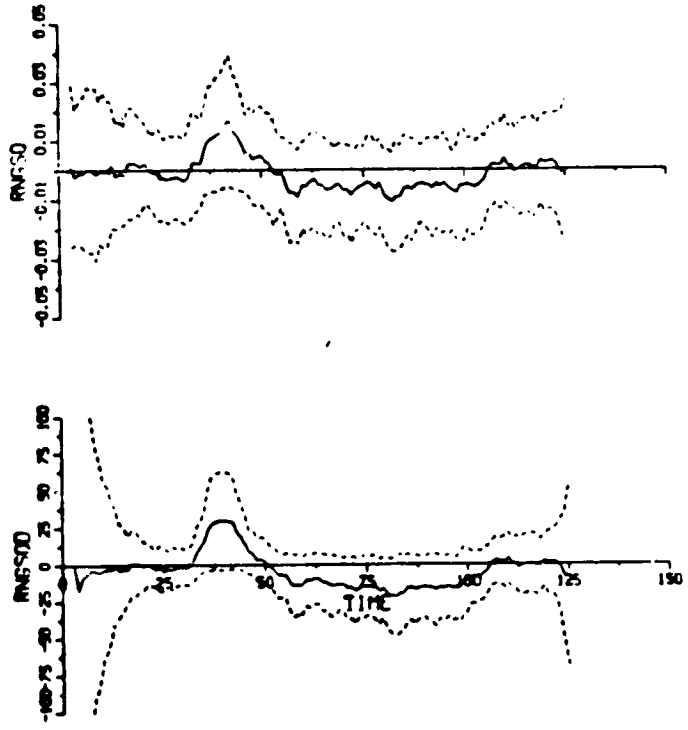
(a) Non-Aided x-y αβ Tracker



(b) Target- and Own-Aided x-y Kalman Filter



(c) Range αβ Tracker



(d) Range Square αβγ Tracker

Figure 31. Statistical Time Plots of Range and Range Rate Errors for the Parallel Turn-in Encounter (twice nominal).

Conclusions

Table 7 shows the summary of the four range estimation algorithms for the various simulation scenarios. The first three cases involve recilinear encounters and the last one involves simultaneous parallel turn-in maneuvers by both aircraft. The following comments apply:

(i) The aided configuration showed the best performance in all cases. The estimation errors are similar in magnitude independent of encounter geometry. The errors are proportional to the input error magnitudes;

(ii) The range and range rate estimates derived from the non-aided x-y filters were better than the other two range axis filters, except the parallel turn-in encounter. The major problem is the large and sustained mean errors caused by the filter dynamic delay;

(iii) The range square filter performed credibly. The low gain nature is apparent in the initial transient error behavior and the dynamic delay errors; and

(iv) The range αβ tracker suffers from the nonlinear range behavior at or near the minimum range, even for rectilinear encounter cases. However, the transient periods are relatively short due to its high gain nature. On the other hand, this high gain nature passes a large portion of the high frequency noise achieving less smoothing than other filters.

Effects of these errors will be analyzed with respect to CAS or CDTI applications in Chapter V. As mentioned earlier, the range and range rate estimates play a very important role in detecting threatening targets.

Table 7. Performance Summary Table of Various Range Estimation Algorithms

Encounter		Non-Aided	Target- and Own-Aided	Range αβ	Range Square αβγ
Tail Chase		43.5(1) 6.8(2)	40.5 6.4	56.6 18.2	49.5 9.0
Route Crossing	1x	42.3 6.7	36.2 5.2	56.0 18.4	49.5 7.5
	2x	94.0 12.5	84.0 9.6	113.6 35.9	94.2 14.1
Head-on	1x	53.9 10.3	44.4 6.8	54.7 18.5	48.4 7.6
	2x	107.7 20.3	87.0 14.3	113.2 36.2	101.0 15.8
	4x	194.6 37.0	173.4 28.5	228.1 73.2	191.6 31.9
Parallel Turn-in	1x	62.9 (91.5)(3) 15.0 (35.1)(4)	39.7 6.1	58.1 21.1 (18.0)	57.8 (81.9) 14.7 (29.2)
	2x	99.8 (92.9) 19.8 (37.3)	80.6 11.7	117.2 38.1 (14.8)	105.1 (109.2) 21.1 (29.2)

(1) steady state standard deviation of range error, in ft.
(2) steady state standard deviation of range rate error, in kt.
(3) maximum mean range error, in ft.
(4) maximum mean range rate error, in kt.

IV

ALTITUDE FILTERS

Because the altitude is the primary axis of separation for the collision avoidance logic, monitoring of vertical state variables is an extremely important function of the TCAS. Therefore, the estimation algorithm is the key element providing the altitude and altitude rate estimates.

Target altitude is decoded from Mode C or Mode S transponder replies. The measured value is a binary integer with the least significant bit representing 100 ft. Figure 32 shows a schematic diagram of the altitude measurement process. According to the current TCAS specifications, Own altitude can be measured in two ways - before or after the Mode C encoding process. Thus, the TCAS processor can access either the transponder altitude data (with the 100 ft quantization) or the TCAS altitude data with much higher resolution. Depending on which of these is used, a proper estimation algorithm needs to be chosen.

The proposed vertical tracker algorithm for TCAS usage is based on the MIT developed Level Occupancy Time (LOT) tracker. It was developed for the active BCAS application in order to overcome the 100 ft quantization of the encoded altitude measurement [13, 14].

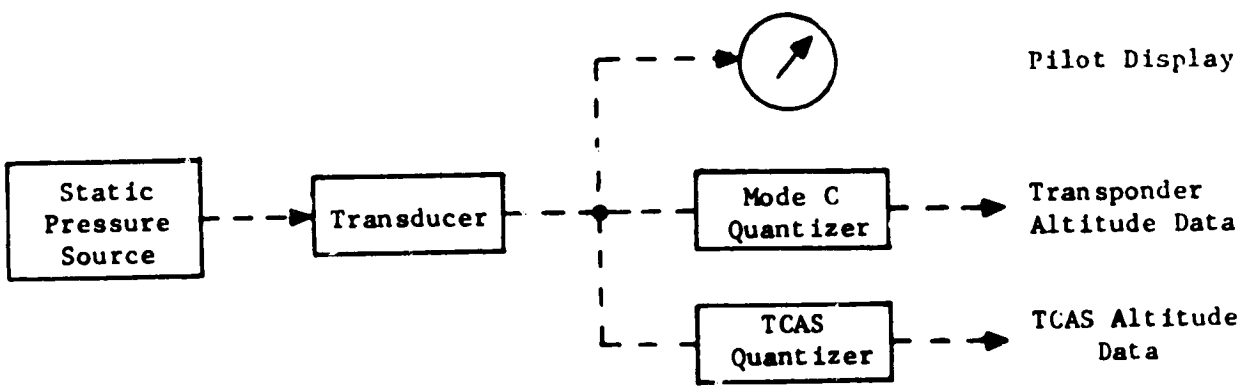


Figure 32. Schematic Diagram of Altitude Measurement Process.

The basic idea of the LOT tracker is to estimate the altitude rate indirectly by estimating the time duration (called the level occupancy time) in a particular quantization level. If the altitude rate is constant, then so is the time duration. Thus, the estimate of level occupancy time, T, is given by

$$\hat{T} = \hat{T} + k\,(T_{meas} - \hat{T}), \quad 0 < k < 1, \tag{52}$$

where

$$T_{meas} = t_{jump} - t_{last\ jump}. \tag{53}$$

Then, the altitude rate estimate is given by

$$\dot{\hat{z}} = 100/\hat{T} \tag{54}$$

Two of the ramifications of the LOT tracker algorithm are:

(1) It requires at least two level changes to obtain rate information; and

(2) It requires at least three level changes to ascertain a change in rate (acceleration).

The vertical tracker software specification contained in the draft TCAS II MOPS is very complex owing to many heuristic logic elements.

In the following sections two algorithms are discussed. These are (1) two-state αβ tracker, and (2) level switching time algorithm.

Alpha Beta Tracker Algorithm

Algorithm The αβ tracker is a linear two-state recursive filter that estimates the aircraft's altitude and its rate of change based upon noise contaminated measurements of altitude. The equations for this algorithm are

$$
\begin{aligned}
&\text{prediction:} && z^{+} = \hat{z}_n + \Delta t \cdot \hat{\dot{z}}_n \\
&\text{altitude error:} && \tilde{z} = z^{m}_{n+1} - z^{+} \\
&\text{altitude update:} && \hat{z}_{n+1} = z^{+} + \alpha\tilde{z} \\
&\text{altitude rate update:} && \hat{\dot{z}}_{n+1} = \hat{\dot{z}}_n + (\beta/\Delta t)\tilde{z} \quad ,
\end{aligned}
\tag{55}
$$

where the measurement, z^m, is given by

$$z^m = z + \xi_z \quad . \tag{56}$$

As usual, when the measurement is missing or invalid because of the surveillance failure, the predicted value is used in place of the measurement. The first two consecutive valid measurements can be used to initialize the estimate as follows:

$$\hat{z}_1 = z^m_1 , \quad \text{and} \quad \hat{\dot{z}}_1 = (z^m_1 - z^m_0)/\Delta t \quad . \tag{57}$$

Gain Selection There are many methods to select the feedback gains. A few methods were discussed in previous chapters. The basic idea is to tune the performance by compromising between the conflicting requirements of a fast response filter and of a good noise smoothing filter. A fast response filter would have a short time constant or wide bandwidth, whereas a good noise smoothing filter would be a sluggish system with long time constant or narrow bandwidth. Thus, it would be natural to end up with a different set of gains depending on different performance measures. For example, by optimizing the error due to "step drift" in velocity, Benedict and Bordner [15] obtained a relationship between α and β ,

$$\beta = \alpha^2/(2-\alpha) \quad . \tag{58}$$

Now, α can be varied to satisfy other filter performance specifications.

The other method is to use the so-called exponential weighted least square error [16,17]. With this criteria, the gain values can be expressed parametrically as

$$\alpha = 1 - \gamma^2 \quad , \qquad (59)$$
$$\beta = (1-\gamma)^2 \quad ,$$

or

$$\beta = 2-\alpha -2\sqrt{1-\alpha} \quad . \qquad (60)$$

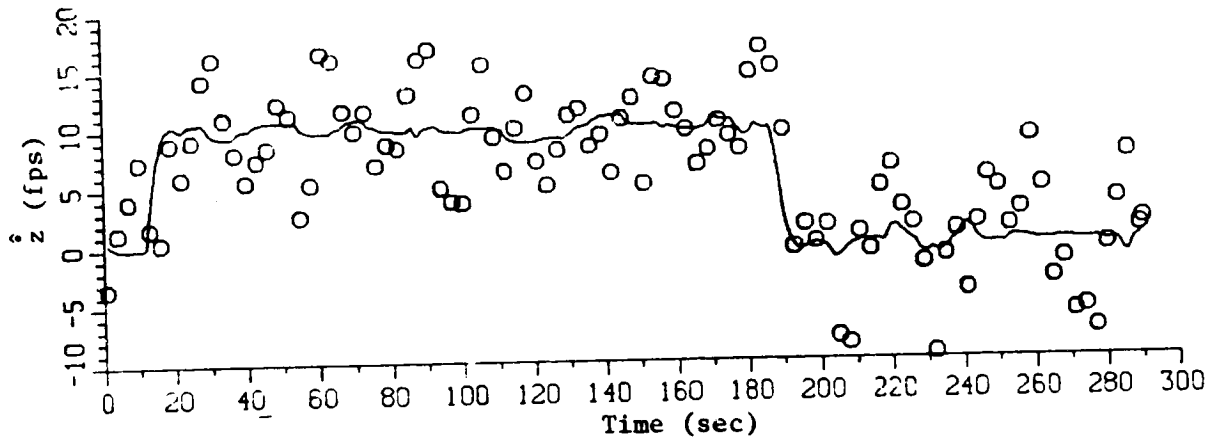
Parameter γ is related to the filter bandwidth ω_b by

$$\gamma = \exp(-\omega_b \Delta t) \quad . \qquad (61)$$

The gain values recommended by the TCAS MOPS are 0.5 and 0.15 for α and β respectively, at the nominal sampling rate of 1 sec.

These gain values are applicable when the altitude measurement error can be modeled by an independent random sequence. When it is no longer independent, as pointed out by the Billmann's study [10], The algorithm itself needs to be modified along the line suggested by the range filter with correlated measurement error. (However, the gain selection problem is a minor one compared to the problem caused by the 100 ft Mode C altitude quantization.)

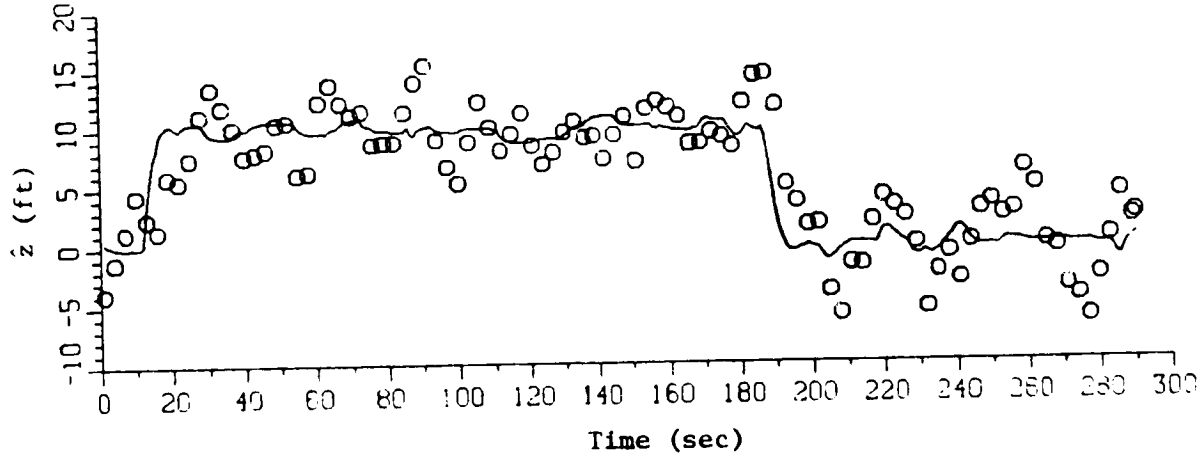
Figure 33 shows sample time plots of altitude rate estimates using two different sets of gains. The results were obtained by the $\alpha\beta$ tracker with altitude measurement containing correlated additive error only. The measurement sequence was obtained by a level-climb (10 fps)-level flight vertical profile. (The experimental set-up will be explained in more detail later.)



(a). α = 0.5 and β = 0.15 Case

True

o o o o o o Estimated



(b) α = 0.486 and β = 0.0803 Case

Figure 33. Altitude Rate Estimates for Different Set of Gains

These examples are used only to illustrate the effect of filter gains. The top plot shows the case with the recommended gains of $\alpha = 0.5$ and $\beta = 0.15$. The bottom plot shows the rate estimate using Eq (59) with $\gamma = 0.717$. For the sample case, the latter is clearly better.

Effect of Quantized Measurements The Traffic Alert and Collision Avoidance System requires estimates of altitude and its rate of change for an intruder aircraft, based upon the Mode C altitude reports. These reports provide the altitude measurements to the nearest 100 ft. If a classical $\alpha\beta$ tracker is used, then a 100 ft "stair case" input induces an undesirable transient response each time the quantization level is crossed.

This can be explained by noting that within an altitude level, the altitude measurement remains unchanged. This causes the estimate of vertical speed to decrease as estimated and measured values of altitude become the same. Figure 34 shows the estimates obtained by the $\alpha\beta$ tracker with Mode C reports as input. It clearly shows the undesirable train of transients. It is noted that the simulation result was obtained using $\alpha = 0.28$ and $\beta = 0.06$. These values are used in the altitude tracker within the surveillance module rather than the CAS logic.

The basic problem with the quantized measurements is the nonlinearity. The apparent error magnitude depends on the actual altitude as well as other additive error sources. The following two examples identify problems caused by quantization.

Example 1: A 100 ft altitude jump occurs when the actual altitude goes from 10.049.9 ft to 10,050.1 ft. This represents an actual rate of + 0.2 ft/sec at the sampling rate of 1 sec. The tracker will process this jump by increasing the rate estimate by β x 100 ft/sec (β is the rate gain). With a typical value of 0.15 for β, this implies that the rate estimate jumps by 15 ft/sec.

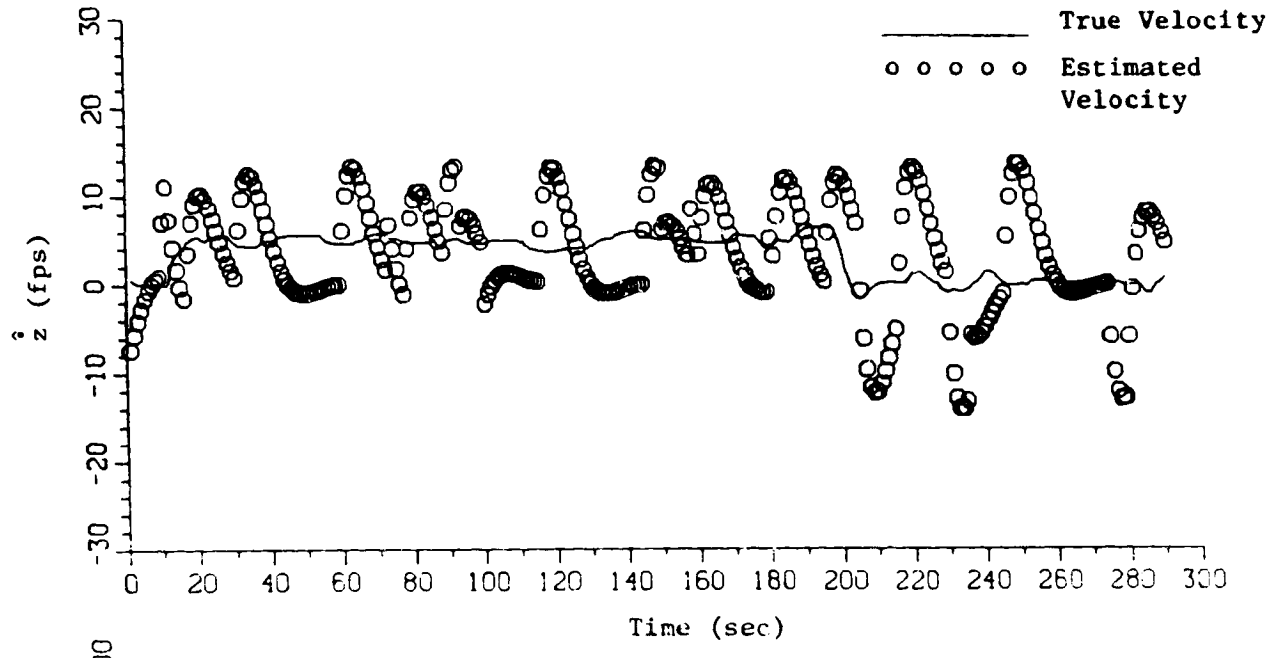
Example 2: When an aircraft is flying at 10,050 ft altitude, then a small magnitude (say 0.2 ft rms) high frequency (random) noise will result in a situation where consecutive measurements are 100 ft apart.

The above mentioned problems indicate that the capability of the αβ tracker in response to quantized measurements is very limited. Alternative algorithms must be considered, which is the subject of the next section.

When the altitude measurement is obtainable with a much finer resolution than 100 ft, then the αβ tracker algorithm would be near optimum*. This is the case with the Own altitude input. Thus, the αβ tracker (with a fine resolution altitude) can be used to judge other filtering schemes in terms of performance.

---

* In many vertical filter implementations for navigation, guidance and flight control applications, other signals are available. These include the attitude angles, body rates and accelerations as well as the barometric altitude. Furthermore, these signals are available for processing at a much higher frequencey (50 msec compared to 1 sec for TCAS). Therefore, the TCAS vertical αβ tracker would not be able to perform as well as other implementations.



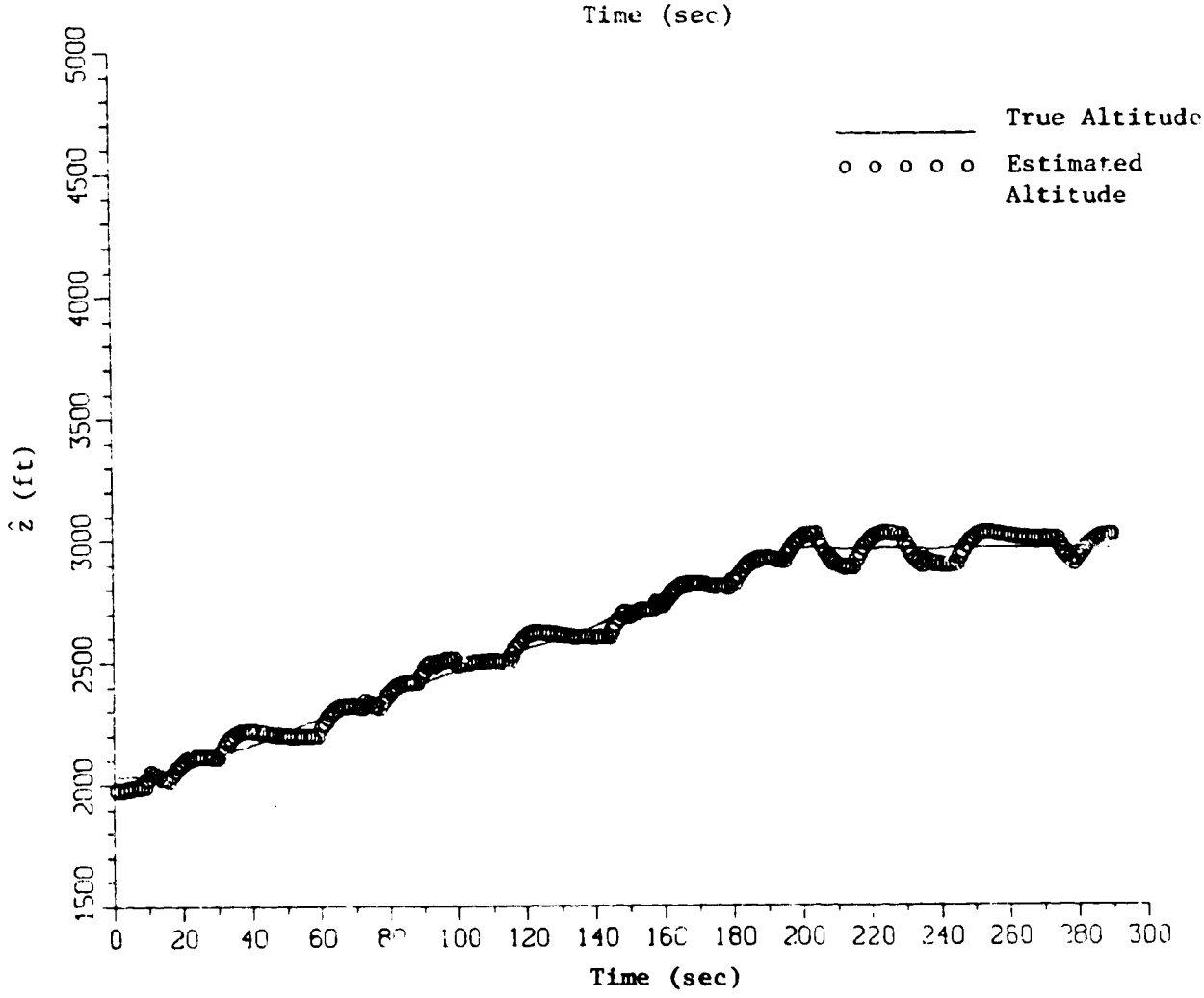


Figure 34. Vertical αβ Tracker Estimates with Mode C Quantized Input.

Level Switching Time Filter

The unsatisfactory response of the αβ tracker to the quantized measurements motivates the search for an alternative algorithm. As mentioned previously, the Level Occupancy Time (LOT) tracker algorithm was designed to fill this void. However, it has very complex software based on heuristic logic. The aim here is to reformulate the problem in a different light.

The basic idea of the level switching time (LST) filter can be summarized in the following steps:

1. Detection of level switching.
2. Estimation of altitude and level switching time.
3. Estimation of altitude and rate based on the last four estimates of altitude and level switching time.
4. Modification of the estimated altitude and rate by error feedback.
5. Validity verification of the modified altitude and rate values.

The following sections explain each of the above steps.

Level Switching Detection Logic The combination of measurement noise and the quantization process can result in an erroneous indication of level change. To prevent such falsely reported level change, the decision on the level switching is made if three out of the past five measurements indicate such a change. This implies that with a sampling time of Δ, it takes at least 3Δ to detect a level change. This delay is necessary to compensate for erroneous level changes caused by noise and the quantization process, and any reduction in this time results in a reliability reduction of the level switching decision.

Estimation of Altitude and Level Switching Time It is clear that more altitude information can be extracted from a segment of reports containing different levels rather than a single level. Consider, for example, a history of Mode C reports

Time	55	56	57	58	59	60	61	62	63	64	(sec)
Report	100	100	100	100	100	100	110	110	110	110	(100 ft).

The segment corresponding to t=56 to 60 sec does not contain any information by itself other than altitude is between 9,050 to 10,050 ft. On the other hand the segment corresponding to t=59 to 63 sec contains a lot more information, viz., the true altitude was 10,050 ft some time during that interval and if the additive noise is small, then that time point would be between t of 60 and 61 sec. Thus, the latter segment is infinitely more useful in pinpointing the true altitude and the point of crossing that altitude. That is, the combination of true altitude and the corresponding time is the important factor.

Motivated by the above argument, the following algorithm is used to estimate altitude and time:

$$z^L = \frac{1}{5}\sum_{i=1}^{m} z_i^c \; ; \quad \text{and} \quad t^L = \frac{\sum_{i=1}^{5} t_i \, z_i^c}{\sum_{i=1}^{5} z_i^c} \; , \tag{62}$$

where z_i^c is the Mode C altitude report times 100 ft at t_i. The four (4) most recent pairs of z^L and t^L are stored in shift registers. These are used to derive a rough estimate of altitude rate discussed next. See Fig. 35 for illustration.

Also stored is the estimated level, L, which is the quantized value of z^L, i.e.,

$$L = \text{Int}[(z^L + 50)/100] \; . \tag{63}$$

If the levels show a cyclic behavior (i.e., the latest level and the one before the previous level are identical), then level flight is declared.

Altitude and Altitude Rate Estimation The stored level switching variables $\{z_k^L, t_k^L) : k=1,..,4\}$ (with k=1 being the latest) are used to generate rough estimates of altitude and altitude rate. The estimates

C2

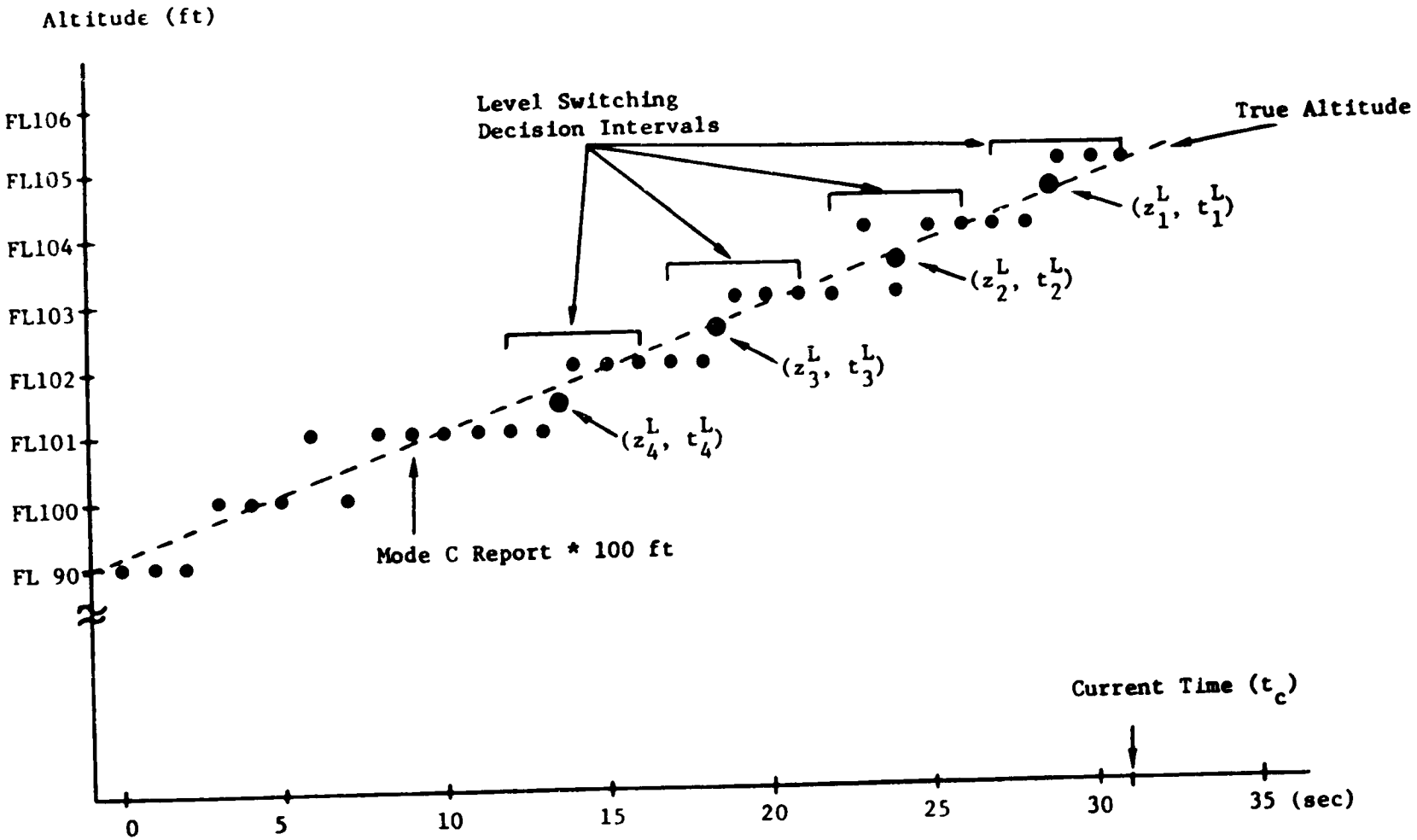


Figure 35. Illustration of Level Switching Times and Altitude Measurements

are obtained using the principal of least squares, assuming that the altitude rate is constant during the time interval of interest. There are four cases to be considered depending on the number of stored measurements.

(i) Altitude Initialization: When only one level switching is observed, then the estimates are initialized as

$$\hat{z} = z_1^L \quad \text{and} \quad \dot{\hat{z}} = 0 \ . \tag{64}$$

(ii) Altitude Rate Initialization: When two level switchings have been observed, then the altitude and rate can be initialized as

$$\hat{z} = z_1^L \quad \text{and} \quad \dot{\hat{z}} = (z_1^L - z_2^L)/(t_1^L - t_2^L) \ . \tag{65}$$

(iii) Three Point Least Squares Fit: When three level switchings have been observed, then the altitude and rate can be estimated as follows. Because of the constant speed assumption, three linear equations can be written at $t=t_1^L$.

$$\begin{aligned} z_1^L &= z + \tilde{z}_1 \ , \\ z_2^L &= z + (t_2^L - t_1^L)\dot{z} + \tilde{z}_2 \ , \text{ and} \\ z_3^L &= z + (t_3^L - t_1^L)\dot{z} + \tilde{z}_3 \ , \end{aligned} \tag{66}$$

where $\tilde{z}_i$ are errors to be minimized. Using the standard least squares method, the estimates at $t = t_1^L$ are given by

$$\begin{bmatrix} \hat{z} \\ \dot{\hat{z}} \end{bmatrix} = \frac{1}{D} \begin{bmatrix} \Delta_2^2 + \Delta_3^2 & -(\Delta_2 + \Delta_3) \\ -(\Delta_2 + \Delta_3) & 3 \end{bmatrix} \begin{bmatrix} z_1^L + z_2^L + z_3^L \\ \Delta_2 z_2^L + \Delta_3 z_3^L \end{bmatrix} \ , \tag{67}$$

where

$$\Delta_2 = t_2^L - t_1^L \ ,$$

$$\Delta_3 = t_3^L - t_1^L \ ,$$

and

$$D = 2(\Delta_2^2 - \Delta_2\Delta_3 + \Delta_3^2) \ .$$

(iv) Four Point Least Squares Fit: When four level switchings have been observed, the previous method can be extended easily. The estimates at $t=t_1^L$ are given by

$$\begin{bmatrix} \hat{z} \\ \dot{\hat{z}} \end{bmatrix} = \frac{1}{D}\begin{bmatrix} \Delta_2^2 + \Delta_3^2 + \Delta_4^2 & -(\Delta_2 + \Delta_3 + \Delta_4) \\ -(\Delta_2 + \Delta_3 + \Delta_4) & 4 \end{bmatrix}\begin{bmatrix} z_1^L + z_2^L + z_3^L + z_4^L \\ \Delta_2 z_2^L + \Delta_3 z_3^L + \Delta_4 z_4^L \end{bmatrix} , \tag{68}$$

where

$$\Delta_i = t_i^L - t_1^L \quad , \quad i = 2,3,4,$$

and

$$D = 4(\Delta_2^2 + \Delta_3^2 + \Delta_4^2) - (\Delta_2 + \Delta_3 + \Delta_4)^2 .$$

It is noted that Δ_i's are "measured" level occupancy times.

The estimates $\hat{z}$ and $\dot{\hat{z}}$ are computed on the basis of the most recent level switching time, t_1^L. The time epoch t_1^L is inevitably at least 2-3 sec behind the current time, t. Thus, the current time estimates need to be extrapolated according to

$$\hat{z}(t) = \hat{z} + (t-t_1^L)\dot{\hat{z}} \tag{69}$$

The altitude rate estimate remains unchanged.

The least squares solution given by Eq (67) or (68) could be recast in a recursive algorithm. However, with the current computer technology, Eq (67) or (68) are straight forward. The recursive form may be advantageous for the least squares fit using six or more data points.

Fine Tuning by Error Feedback The previous methods apply only when a level switching is detected by the "3 out of 5" rule. At the time of detection, a new level switching time, t_1^L, and the corresponding altitude, z_1^L, are available. This time t_1^L could be very much behind the current time. Thus, there could be a big time gap between the estimate updates.

The time interval during which the Mode C altitude reports indicate the same flight level provide "passive" information. The information will not allow us to compute the estimates, but it provides a means to check if the available estimates are still reasonable. This can be achieved by comparing the "predicted" altitude at the current time (given by Eq (69)) with the current Mode C altitude. If the difference between the two is less than 50 ft (a half of Mode C quantization) then the last estimates seem to be still accurate. On the other hand, a difference of more than 50 ft indicates (a) the last estimates are in error; (b) the target dynamics has changed; or (c) the Mode C report is in error due to noise. The estimates need to be updated in cases (a) and (b) but not in case (c). This suggests that the average difference over some time period should be used rather than instantaneous difference.

To accomplish this, the average value of error for the past five sampling times is calculated. Any amount in excess of $\pm$ 50 ft is fedback to modify $\hat{z}$ and $\dot{\hat{z}}$. Therefore, $\hat{z}$ and $\dot{\hat{z}}$ are updated according to the following equations:

$$(\hat{z})_{New} = (\hat{z})_{Old} + k_1 \tilde{z} \quad , \tag{70}$$

$$(\dot{\hat{z}})_{New} = (\dot{\hat{z}})_{Old} + k_2 \tilde{z} \quad ,$$

where

$$\tilde{z} = \begin{cases} \bar{z} - 50 \ , & \text{if} \quad \bar{z} > 50 \\ \bar{z} + 50 \ , & \text{if} \quad \bar{z} < -50 \end{cases} \tag{71}$$

and

$$\bar{z} = \frac{1}{5} \sum_{i=1}^{5} \left(z_i^c - \hat{z}_i \right) . \tag{72}$$

The $\hat{z}_i^c$'s are the Mode C reported altitudes (times 100 ft) at the five most recent valid reporting times, t_i. (The reports and times are stored in five tier shift registers and used in the level switching detection logic.) The z_i are predicted altitudes extrapolated from the last level switching time, t_1^L. Thus,

$$\hat{z}_i = \hat{z}_{Old} + (t_i - t_1^L) \cdot \dot{\hat{z}}_{Old} \ . \tag{73}$$

k_1 and k_2 are the feed back gains for the altitude and rate update The values of k_1 and k_2 must be selected by compromising between the requirements for rapid modification and possible instability (divergence of $\hat{z}_i$'s) caused by large values. Values for k_1 of 0.632 and k_2 of 0.155 are reasonable values as will be shown in a later section.

Validity of the Modified Altitude and Rate Values To verify validity, the modified values of $\hat{z}$ and $\dot{\hat{z}}$ are used to recompute the estimates of altitude at the last five sampling times, and the average error $\bar{z}$ is recomputed. If the absolute value of the resulting average error is less than 50 ft, the validity of the modified values is established. To avoid excessive modification of altitude, this modification is limited to $\pm$ 35 ft, i.e., the modified estimate $\hat{z}_{New}$ in Eq (70) would not be changed by more than 35 ft from the originally computed $\hat{z}_{Old}$ by the least squares algorithm.

Figure 38 shows an over-all computational flow of the proposed Level Switching Time filter.

Remark A basic ingredient of the proposed tracker algorithm is the idea of obtaining the rough altitude and altitude rate estimates based on the estimated altitude and level switching time (z^L, t^L) pairs. The three-out-of-five rule was specifically designed for the sampling period of 1 sec. The same rule may not apply or be desirable for other sampling periods. Therefore, other methods of determining the (z^L, t^L) pair must be devised. However, the rest of the algorithm should be applicable without modifications. The modular construction of the logic would ease the task of replacing the level switching time detection logic.

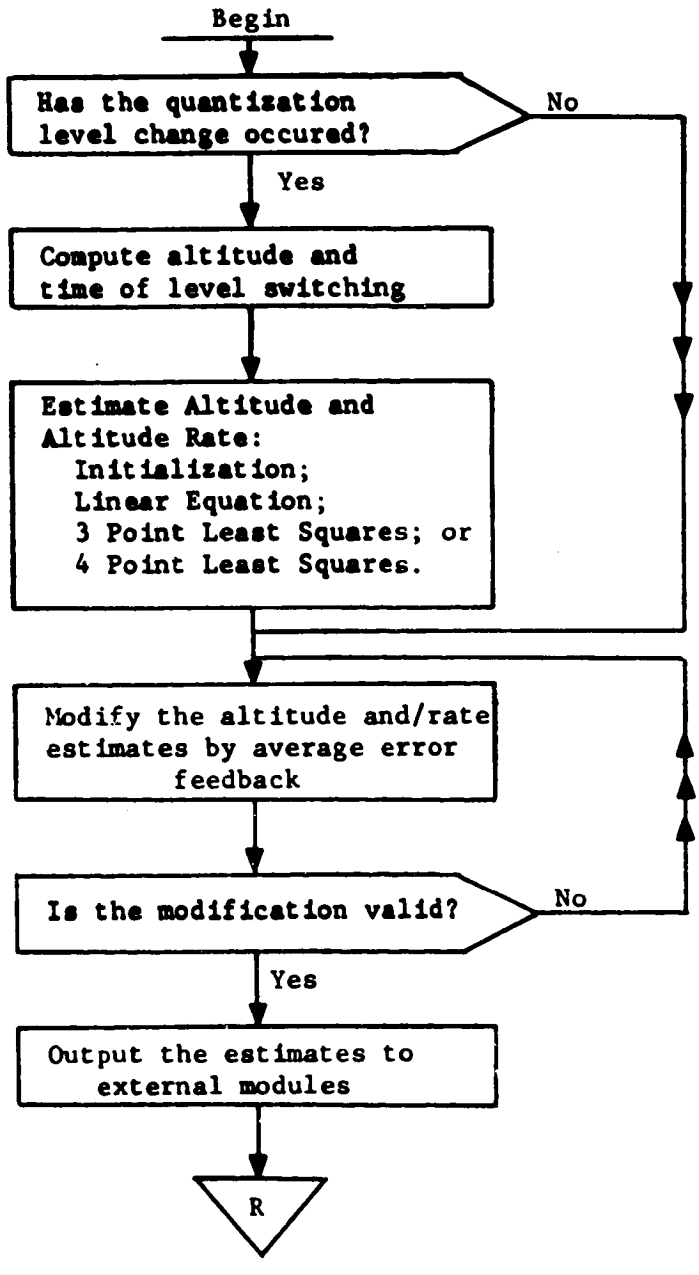


Figure 36. Level Switching Time Filter Macro Flow Chart.

Monte Carlo Simulation Results

This section describes the simulation set up, and it analyzes the proposed Level switching Time (LST) filter algorithm as well as the αβ tracker. The αβ tracker algorithm with non-quantized altitude input will be the basis of comparison.

Simulation Set Up The aircraft vertical dynamics are simulated by a simple second order altitude select/hold logic. In equation form

$$\frac{d}{dt} z = \dot{z} \quad , \qquad \frac{d}{dt} \dot{z} = \ddot{z}_c \quad , \tag{74}$$

where the acceleration command, $\ddot{z}_c$, is given by

$$\ddot{z}_c = [[\frac{1}{\tau_v} (\dot{z}_c - \dot{z} + \xi_{\dot{z}})]]_{\ddot{z}_{ref}} \quad . \tag{75}$$

Also, the velocity command, $\dot{z}_c$, is given by

$$\dot{z}_c = [[\frac{1}{\tau_z} (z_{ref} - z)]]_{\dot{z}_{ref}} \quad . \tag{76}$$

In the above expressions the term $\xi_{\dot{z}}$ is the altitude rate noise injected to simulate such phenomena as wind gusts and pilot activities. It is designed to change magnitude on a 20 sec average basis, with a standard deviation of 0.75 fps. The notation $[[x]]_y$ means x is authority-limited to y. z_{ref}, $\dot{z}_{ref}$ and $\ddot{z}_{ref}$ are the altitude select references, i.e., they are the desired altitude, maximum rate magnitude and maximum allowable acceleration. Once z_{ref} is selected, the vertical velocity command, $\dot{z}_c$, is generated by Eq (76). If the current altitude is not close to the reference altitude, the command takes on the maximum value, $\dot{z}_{ref}$. If the altitude is close to the reference, then the command takes a smaller value. The acceleration command, $\ddot{z}_c$, is generated in a similar manner depending on the (previously computed) velocity command, $\dot{z}_c$, and the current velocity state value, $\dot{z}$.

For example, if the current velocity is close to the velocity command within the pilot noise input, then the acceleration command takes on a small value. Therefore, by selecting values for z_{ref} and $\dot{z}_{ref}$ ($\ddot{z}_{ref}$ is fixed to 2.5 fpss), reasonable altitude and altitude rate time histories can be generated. In the simulation program, Eqs(74)-(76) are computed at an integration cycle of 0.2 sec using a trapezoidal integration routine.

The altitude generated by the above method is sampled at a one second interval. High frequency additive noise is added to the true altitude which in turn is quantized to the nearest 100 ft. Figure 37 shows the overall block diagram of this process.

The additive error model is generated by a second order correlation process* as suggested by Billmann [10]. The generating equation is given by

$$\tilde{z}_n = 1.066\, \tilde{z}_{n-1} - 0.191\, \tilde{z}_{n-2} + \xi_{z,n} \quad . \tag{77}$$

Here, $\xi_{z,n}$ is a random noise with a normal density function, mean value of zero, and standard deviation of 10.5 ft. By solving the associated Yule-Walker equation [18], it can be shown that the steady state standard deviation of the additive error is 23.8 ft. The measured altitude is given by

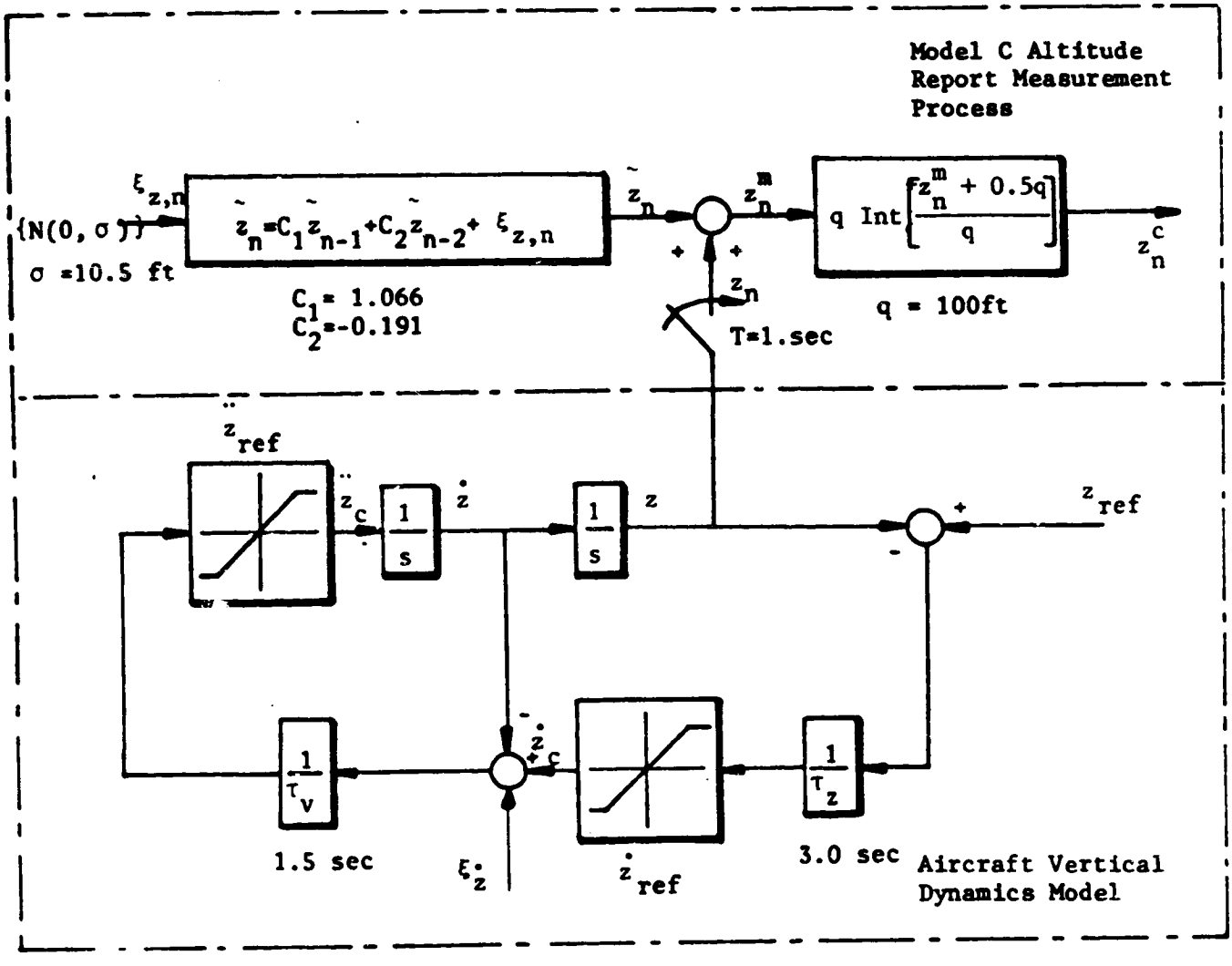
$$z_n^m = z_n + \tilde{z}_n \tag{78}$$

represents the baro-altimeter output, and it is input to the $\alpha\beta$ tracker for generating Own altitude estimates. It is also input to generate the Mode C reported altitude.

The Mode C quantized altitude, z_n^c, is given by the equation

---

* At a preliminary simulation stage, it was found that the level switching detection logic was sensitive to the nature of additive noise. For example, a two-out-of-three rule worked reliably with white noise error [illegible] it did not work for correlated noise. The three-out-of-five [illegible] works for both types of error.



z_{ref} = desired altitude

$\dot{z}_{ref}$ = desired altitude rate

$\ddot{z}_{ref}$ = maximum vertical acceleration authority limit (2.5 ft/sec^2)

Figure 37. Aircraft Vertical Dynamic and Measurement Process Model

$$z_n^c = q \cdot \text{Int}\left[\frac{z_n + \dot{z}_n + 0.5\,q}{q}\right] \tag{79}$$

where q is 100 ft and Int[x] means the integer value of x.

Figure 38 shows time plots of true altitude rate, altitude and the Mode C altitude reports (plotted every 3 sec) for the case of a pull-up maneuver of 2000 to 3700 ft at a rate of 10 fps. It is noted that the rate shows a small magnitude execursion about the command value of 10 fps.

The αβ tracker and LSF tracker algorithms, together with the measurement generation module, were assembled together in a sixty pass Monte Carlo simulation program to generate statistical performance data. The true altitude and altitude rate time histories are computed once in the first pass. Afterwards, a new set of altitude errors are added to the true altitude which is fed into the tracker algorithms.

Simulation Scenarios Several different simulation scenarios were examined during the course of this study. In all thirteen cases were run with the altitude profiles generated for vertical rates of ± 5, 10, 20, and ± 60 fps. Most of these are run to obtain raw statistical performance data. Some of these are run in conjunction with others to obtain "sensitivity" data. Therefore, the total number of computer runs made during the study was considerable including the algorithm development, design and tuning stages.

The following cases are discussed:

1. Effect of quantization on the αβ tracker;
2. Effect of α and β gain variations;
3. Performance change due to difference in additive errors; and
4. Selected individual cases.

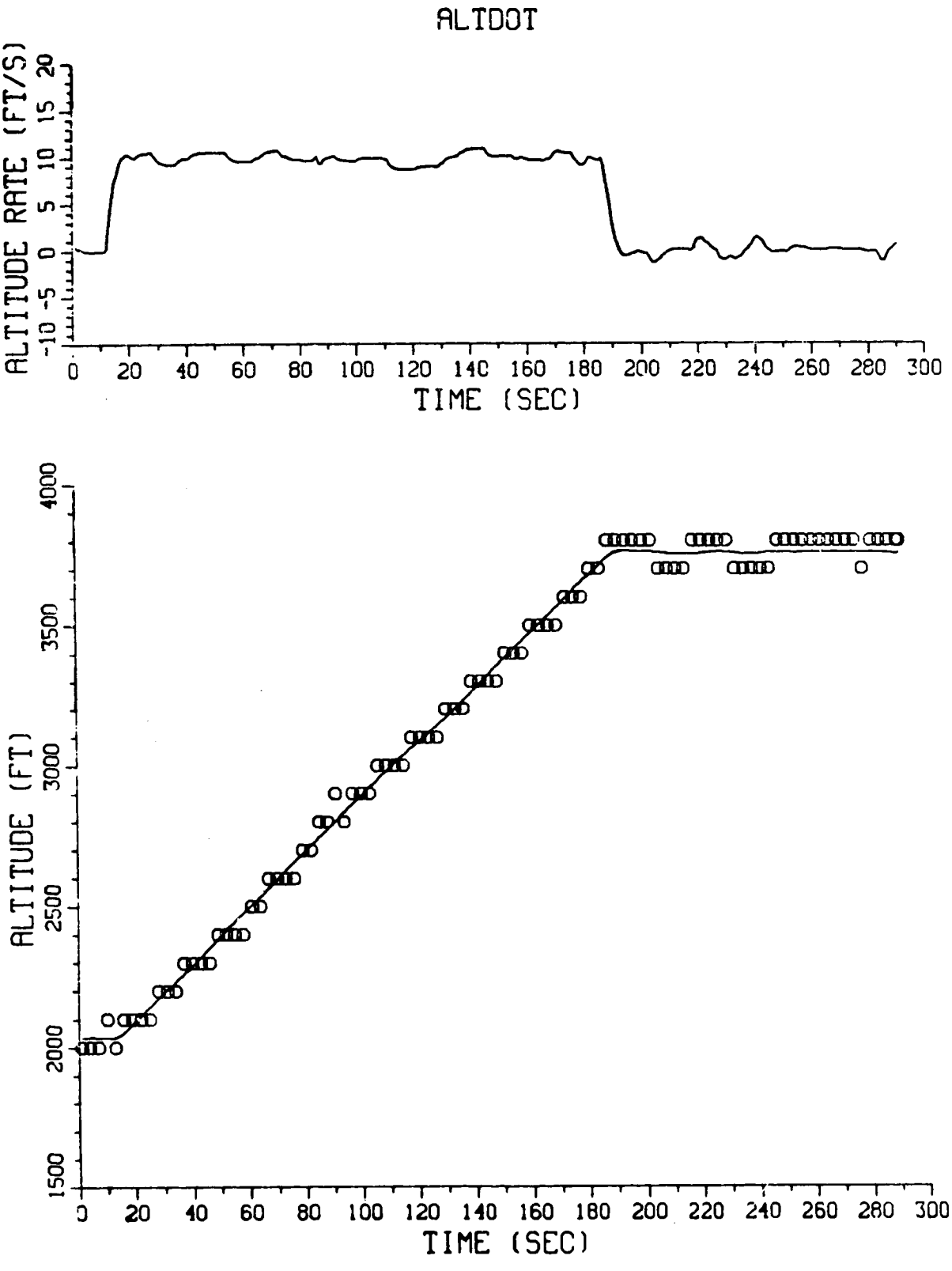


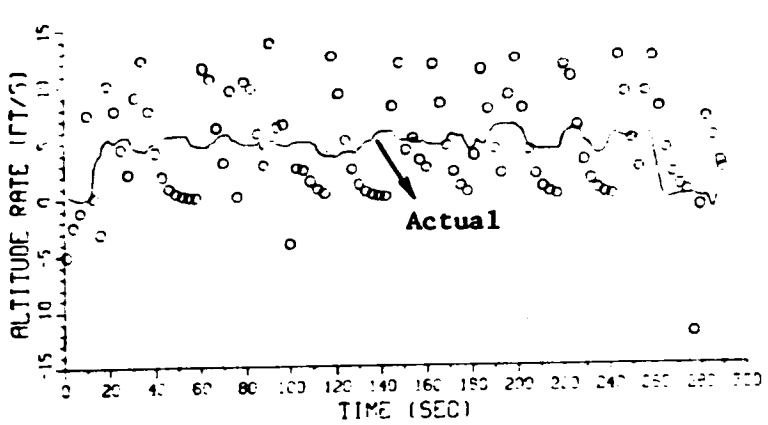
Figure 38. Time Plots of Vertical Rate, Altitude and Mode C Reported Altitude.

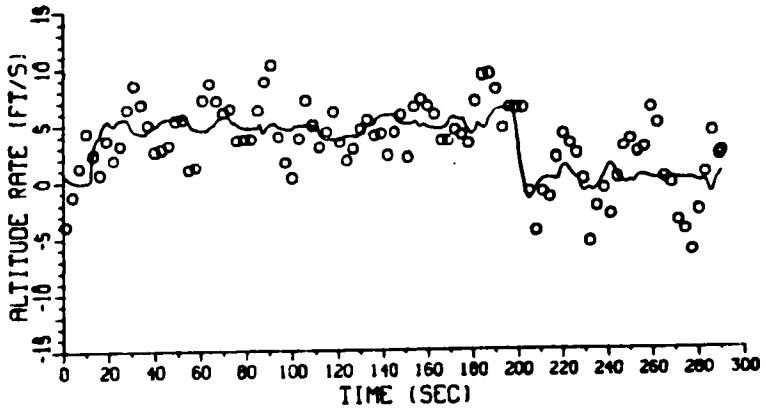
Effect of Altitude Quantization in the αβ Tracker Algorithm Besides its use on the high resolution Own altitude input, the αβ tracker may be used as a preprocessor in the surveillance function of correlating targets with the gating technique discussed previously. The algorithm is used with the Mode C reported altitude. Therefore, it is important to obtain a basic idea of how it performs under such circumstances.

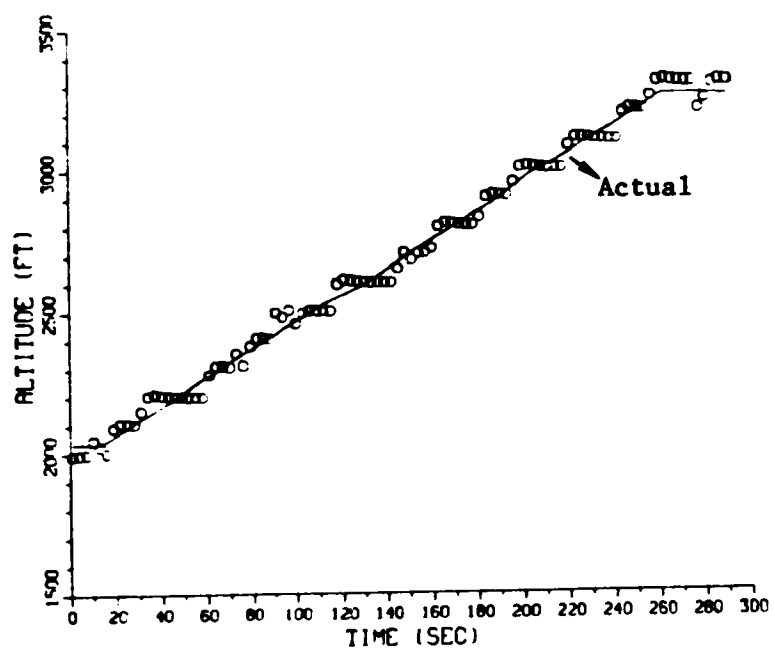
Quantization values of 25 ft and the regular Mode C 100 ft were tested. The quantization of 25 ft represents two more bits of altitude information in the Mode C transponder reply. Figure 39 shows sample altitude and altitude rate time plots for a 5 fps climb profile. The estimates are plotted every 3 seconds. Clearly, the quantization size has a large effect. For the 100 ft quantization case, the altitude estimate shows the stair-step characteristics. The rate estimate shows a sequence of transients, the estimate jumps to large values and then decays to zero and so forth. For the 25 ft quantization case, the altitude estimate does not show the stair-step characteristics. It follows the true altitude fairly smoothly. The rate estimate tends to follow the true value more faithfully. The transient effect is still very much in evidence. This can be seen when the aircraft levels-off. In this phase, the rate estimate is jumping around. This is caused by the tendency for the measurements to change quantization levels more frequently. This indicates that even though the estimates are improved when the quantization is 25 ft, the classical αβ tracker algorithm may not provide sufficiently accurate estimates. The following Table 8 summarizes the error statistics in terms of average rms. (Mean and standard deviation values do not mean very much when the estimates contain so much transient behavior.)

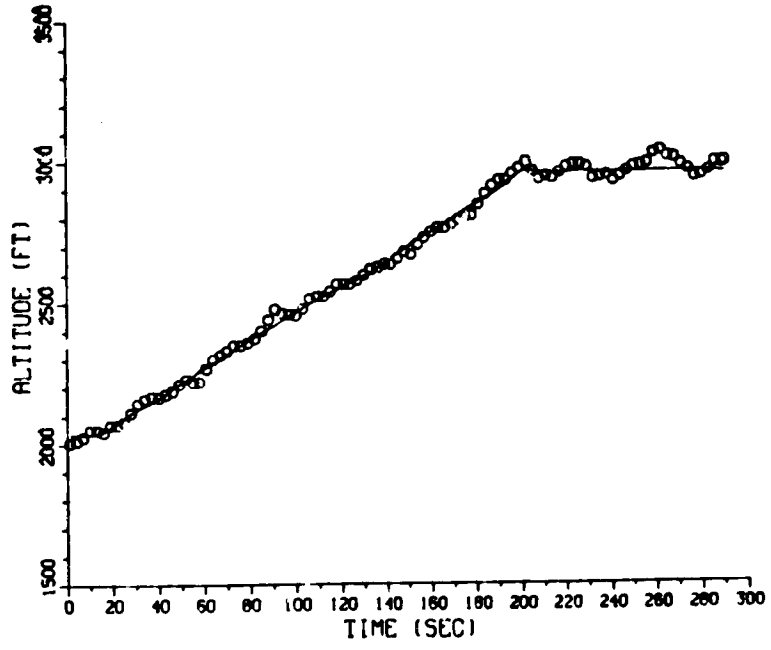
Table 8. Quantization Level Effects on Error Magnitudes

Quantization (ft)	100	25
Altitude error rms (ft)	35	25
Altitude rate error rms (fps)	5.3	4.3









(a) 100 ft Quantization

(b) 25 ft Quantization

Figure 39. Quantization Effect on the αβ Tracker Estimates

Effect of the αβ Filter Gains The effect of filter gains is examined by comparing high and low gain configurations. The previously described correlated error model and a 10 fps climb altitude profile were used for simulation comparison. The high gain configuration used the α and β values of 0.5 and 0.15 respectively. This set is the recommended values for obtaining Own altitude and altitude rate estimates within the CAS logic. The low gain configuration uses the α and β values of 0.486 and 0.0803, respectively. These gains are derived by the exponential weighting method. See Eq (59). The altitude measurements are not quantized.

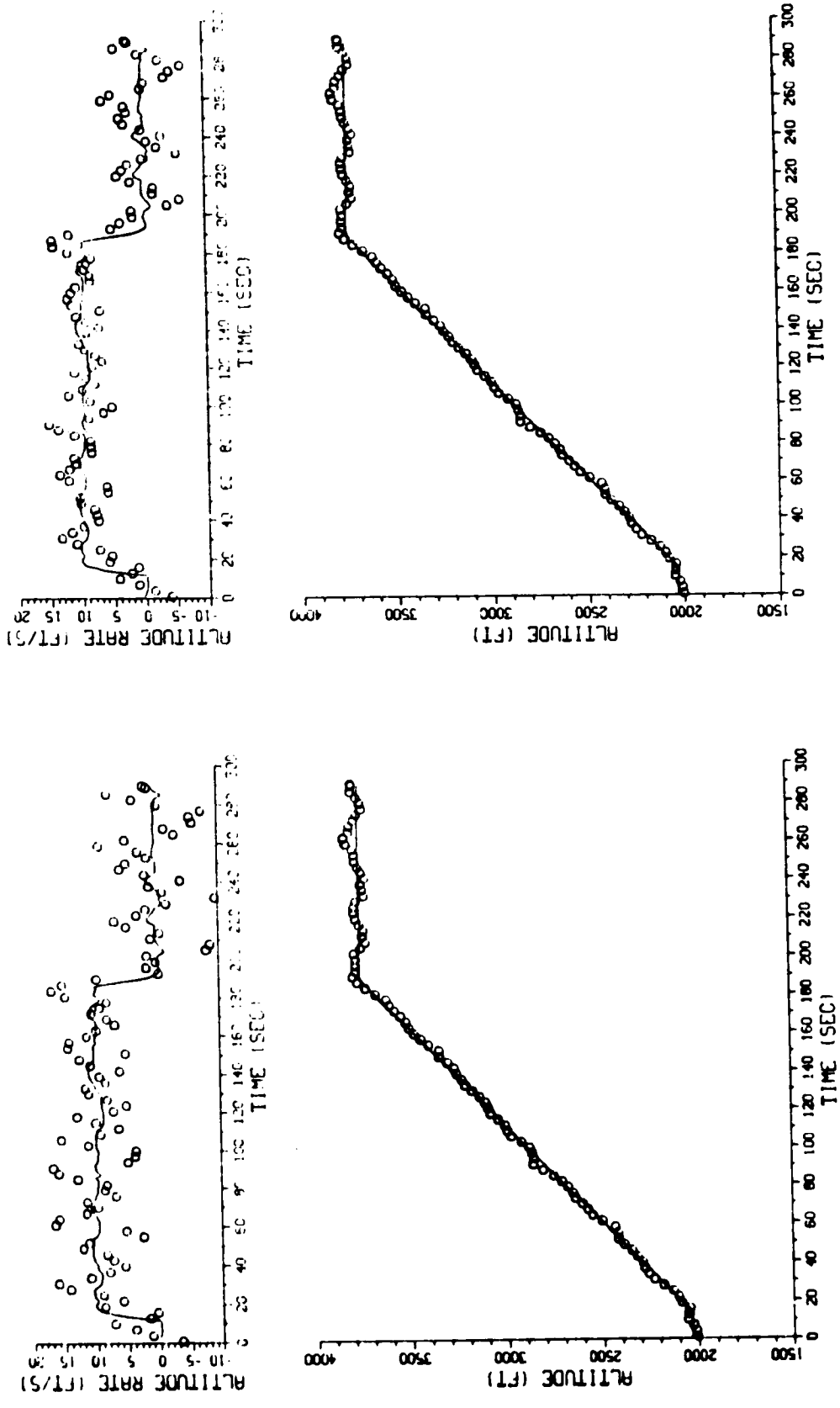
Figure 40 shows sample altitude and altitude rate estimate time plots. The altitude estimates are very similar, reflecting the fact that the proportional (α) gains are practically equal. The altitude rate estimates are substantially different. The high gain configuration shows much larger errors compared to the low gain configuration.

The reason for using a high rate gain (β) is to track the rate during the acceleration and deceleration phases, i.e., it provides a wider bandwidth. From the simulation results this point is not apparent. The low gain configuration does not show a particularly sluggish response when the altitude levels off. The mean peak error during this period is a little larger for the latter configuration, however.

Table 9 summarizes the steady state error statistics in terms of error standard deviations.

Table 9. Gain Effects on Error Magnitudes

	High Gain $\alpha = .5$ $\beta = .15$	Low Gain $\alpha = .486$ $\beta = .0803$
Altitude error deviation, ft	35	35
Altitude rate error deviation, ft/sec	5.5	3.5



(a) High Gain αβ Tracker

(b) Low Gain αβ Tracker

Figure 40. Effect of Filter Gains on the αβ Tracker

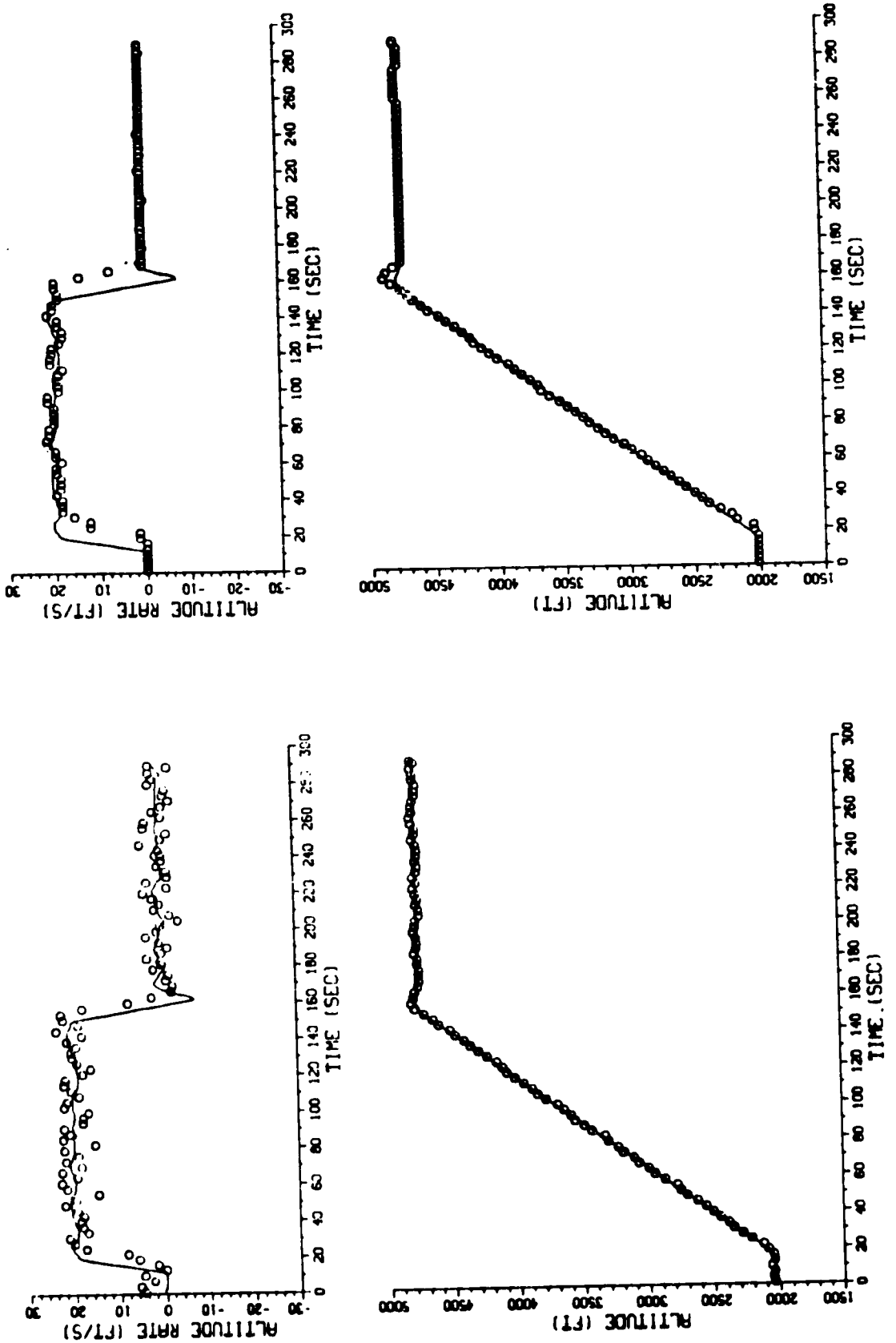
Different Error Characteristics The purpose of this section is two-fold. One is to examine the effect of additive noise which is injected to measurement prior to the quantization process. The other is to compare the performance of the αβ tracker (without quantization) and the LOS tracker (with the standard 100 ft quantization).

The error variations are (1) zero mean white noise with the standard deviation of 23.8 ft, (2) the correlated noise identified by Billmann (Eq (77)), and (3) the same noise but with 95% report validity rate. The standard deviation of 23.8 ft for the white noise was chosen to match the steady state standard deviation of the correlated noise. The report validity rate is defined to be the probability of receiving a valid reply to an interrogation transmission. Thus, it means that 5% of the time, the Mode C reported altitude is missing. The 95% report validity rate is achievable with the current TCAS II design [19].

The flight scenario is a 20 fps climb profile from 2000 (FL20) to 4700 ft (FL47). Figures 41 and 42 show sample and error statistics time plots respectively for the white noise case. The αβ tracker results are on the left and the LOS tracker results are on the right. Figures 43 and 44 show the similar quantities for the correlated noise case and Figs. 45 and 46 show results by using correlated noise and 95% report reliability. Table 10 shows the summary of the average statistics in comparing these cases.

The following comments and remarks are derived from the simulation results.

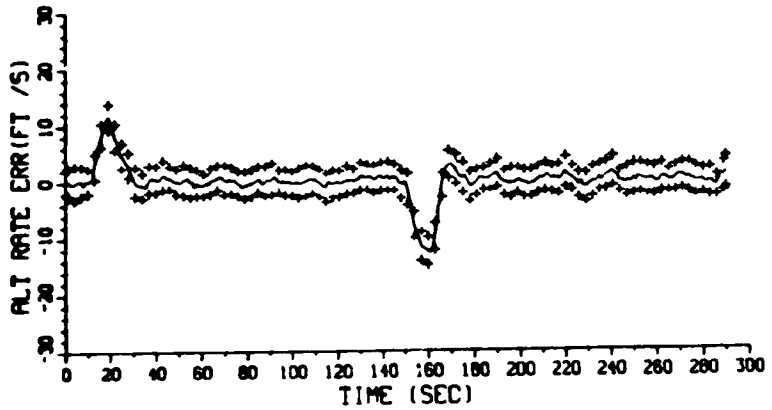
(i) As previously stated, the high frequency estimation errors (as shown by the standard deviations) are smallest for the white noise additive error case. For the αβ tracker, the explanation is that this filter is based on the white noise assumption but not on the time correlation model. A most likely explanation for the LST tracker is that the level switching detection logic utilizes the sum of five measurements. In this case, independent errors tend to cancel each other.

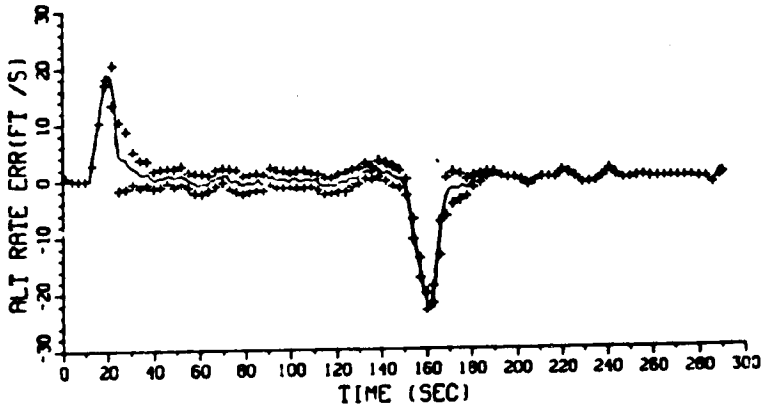


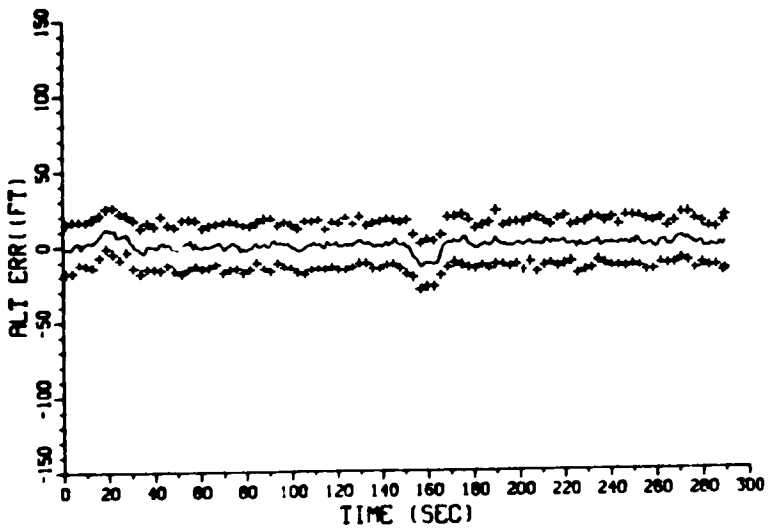
(a) αβ Tracker

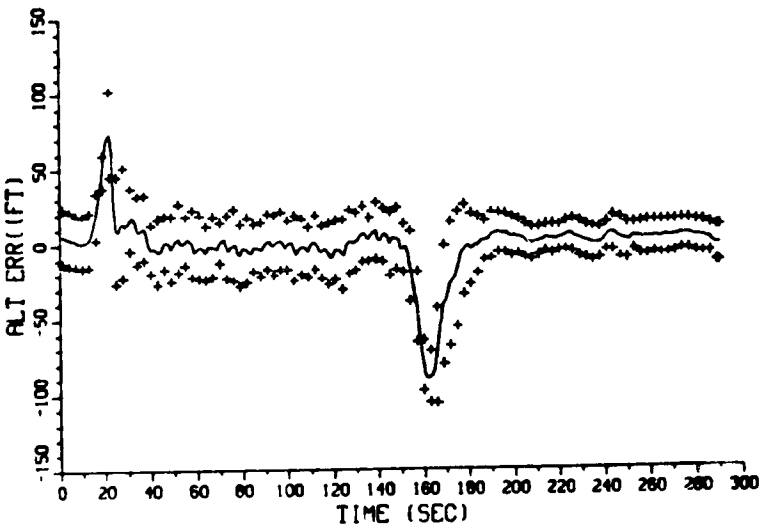
(b) LST Tracker

Figure 41. White Noise Additive Error - Sample Time Plots





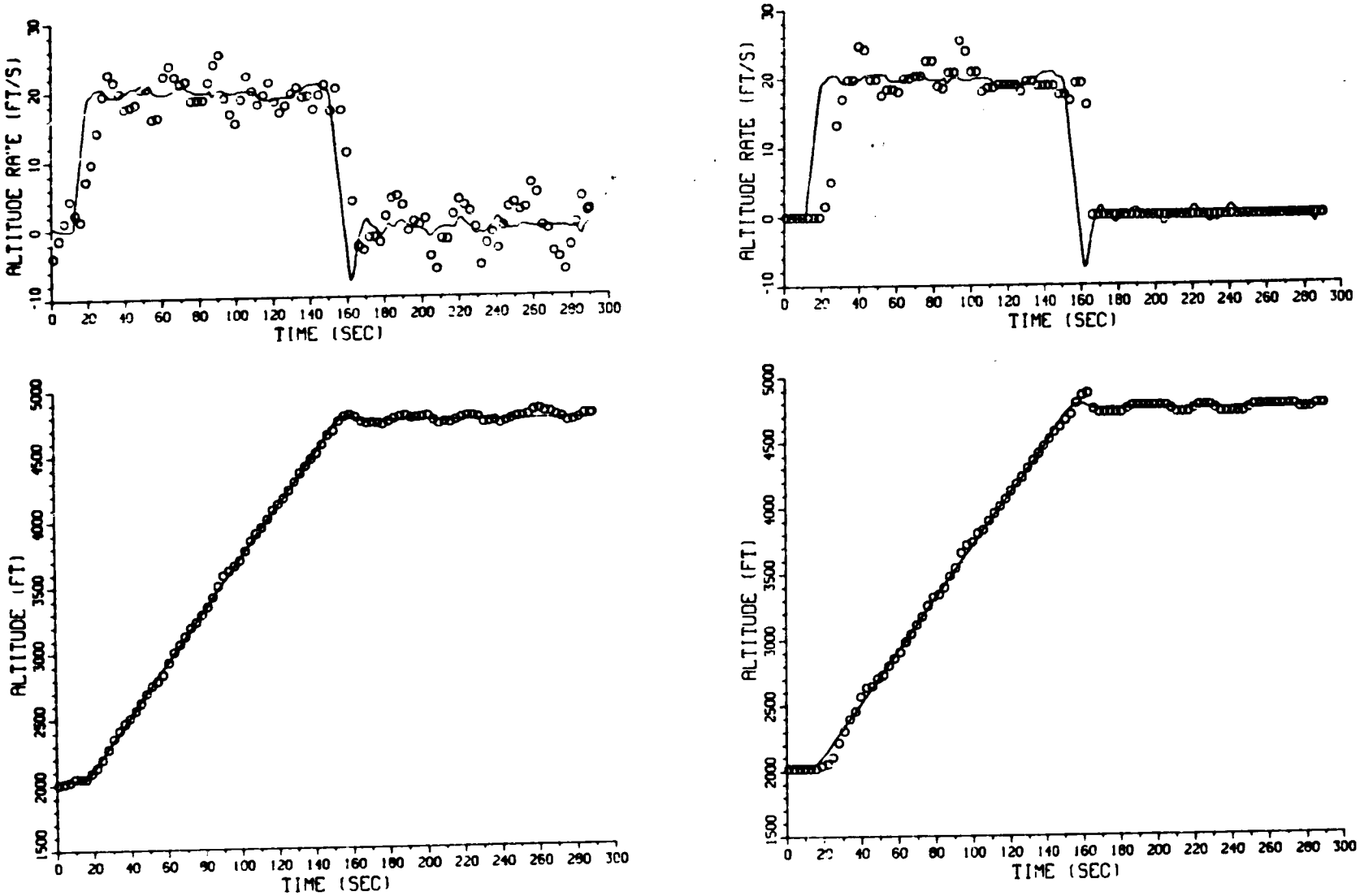




(a) αβ Tracker

(b) LST Tracker

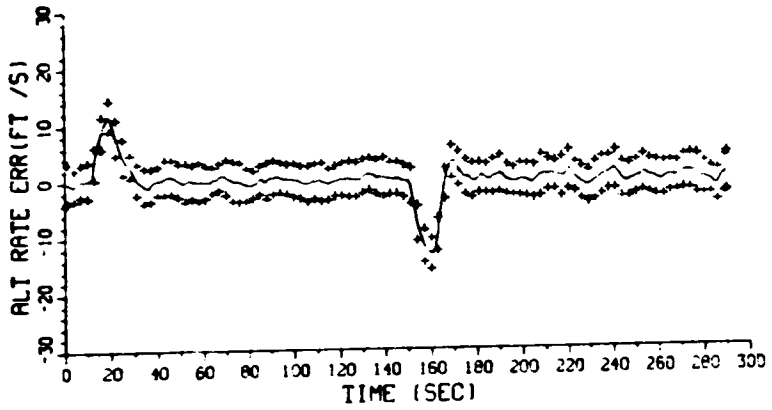
Figure 42. White Noise Additive Error - Statistical Error Plots

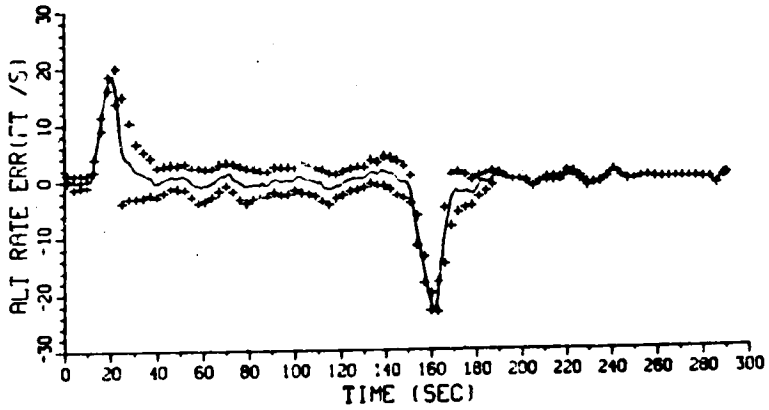


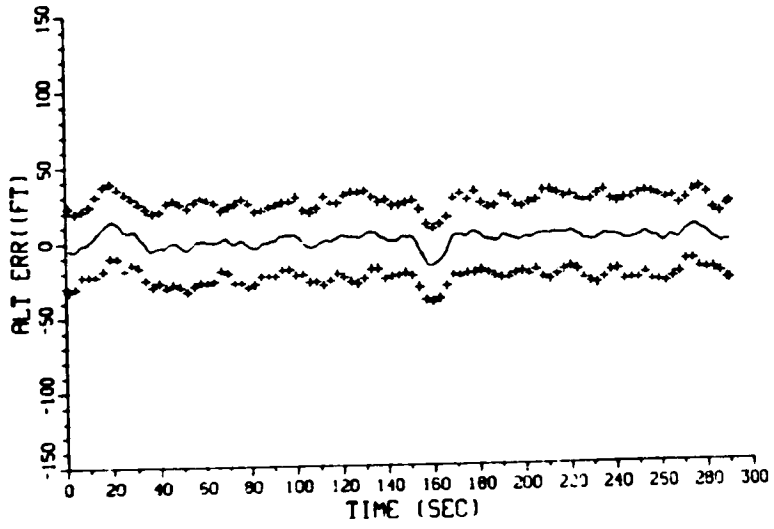
(a) αβ Tracker

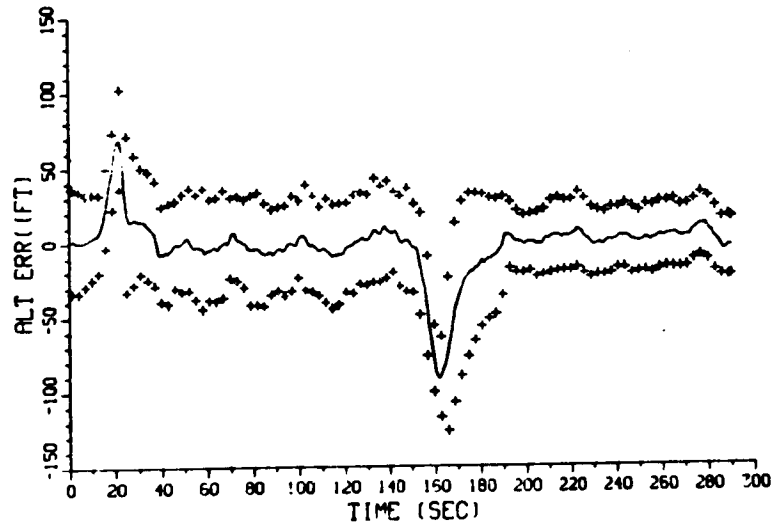
(b) LST Tracker

Figure 43. Correlated Noise Additive Error - Sample Time Plots





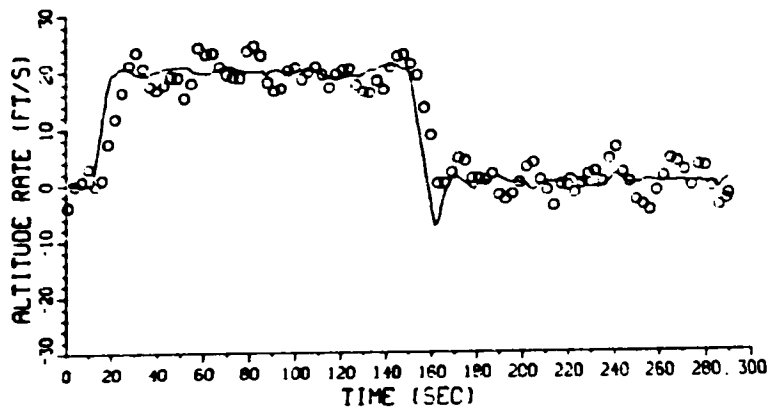


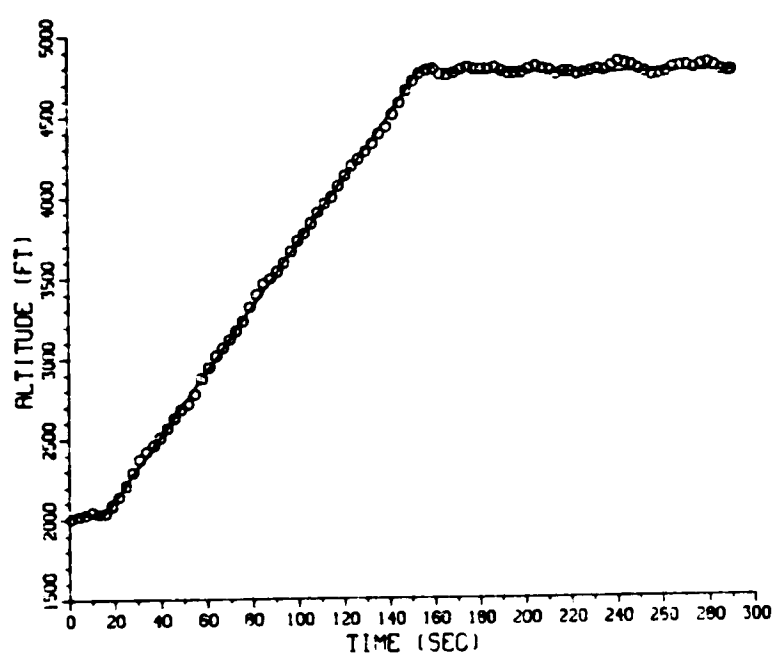


(a) αβ Tracker

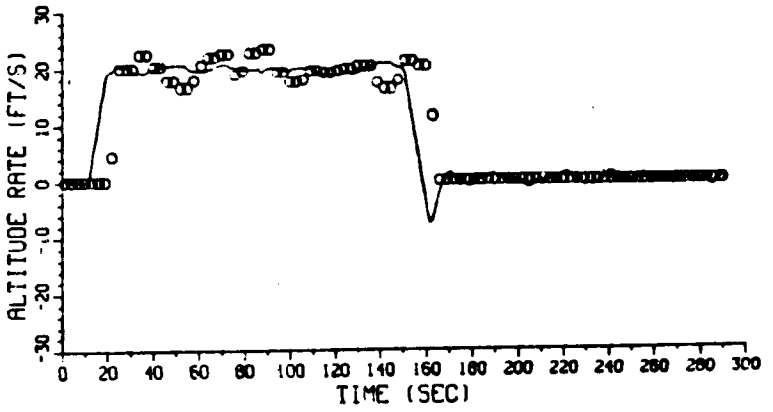
(b) LST Tracker

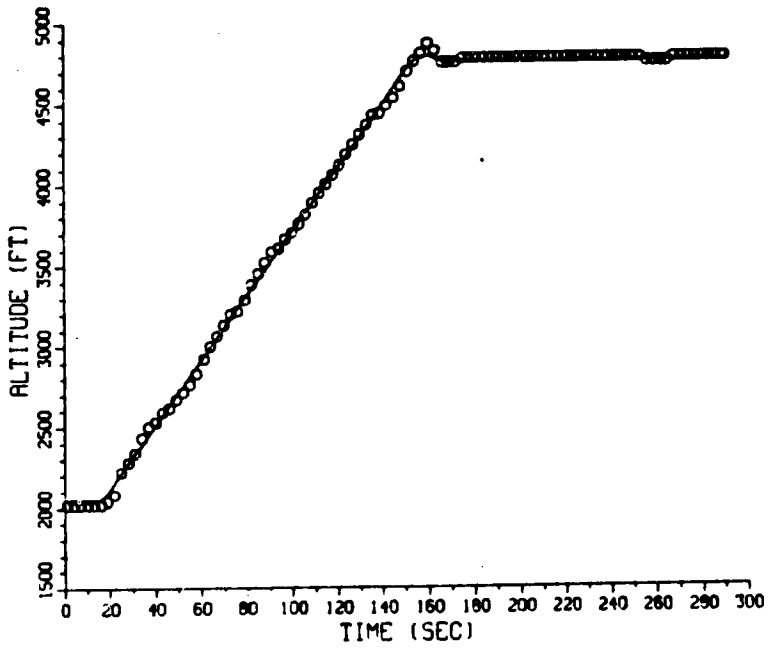
Figure 44. Correlated Noise Additive Error - Statistical Error Plots





(a) αβ Tracker





(b) LST Tracker

Figure 45. Correlated Noise Additive Error with 95% Reliability - Sample Time Plots

(a) αβ Tracker (b) LST Tracker

Figure 46. Correlated Noise Additive Error with 95% Reliability - Statistical Error Plots

Table 10. Statistical Summary of Additive Noise Error Effect

	αβ Tracker	LST Tracker
Wh. e Noise	15.5 (13.8)*	16.7 (85.7)
	2.5 (12.9)	1.6 (20.8)
Correlated Noise	28.9 (23.5)	31.2 (83.9)
	3.0 (11.9)	2.4 (20.8)
Correlated Noise with 55% Reliability	28.7 (18.5)	34.9 (85.4)
	4.6 (12.3)	3.3 (20.5)

* Altitude standard deviation (ft) Peak mean error (ft)
Altitude rate standard deviation (FPS) Peak mean error (fps)

(ii) The mean errors for the LST tracker are very similar for the three cases. This indicates that the noise input has a secondary effect compared to the primary effect of the 100 ft resolution. The mean errors for the αβ tracker are similar except for the altitude error for the white noise case which is smaller.

(iii) The steady state altitude rate errors for the LST tracker are better than those of the αβ tracker. The average improvement is 28%. During level light, the LST tracker is locked on to the nominal 0 fps. All of the LST tracker errors in this region reflect the rate noise injected to the altitude profile generation model. The altitude rate errors for the αβ tracker are affected by the high frequency measurement noise.

(iv) The αβ tracker attenuates the white noise magnitude to 66% in the position estimate (15.5 vs. 23.8 ft). This is not the case with the correlated noise. In fact, the noise magnitude is amplified by 21% (28.8 vs. 23.8 ft). This implies that the filter is not tuned for the colored noise.

(v) The high frequency altitude errors for the LST tracker are larger by only 12% compared to the αβ tracker. This is despite the 100 ft quantization used for the LST algorithm.

(vi) The average peak errors for the αβ tracker are 18.6 ft and 12.4 fps for the altitude and rate. The comparable numbers for the LST tracker are 85 ft and 20.7 fps. Essentially the LST tracker errors are induced by pure time delay corresponding roughly to 100/z.

(vii) The dynamic delay times were 12.9 sec for the αβ tracker and 15.7 sec for the LST tracker. Here the dynamic delay time is measured by the average time interval between the 4 fps error point in the mean altitude rate error history. The delay is thought to be equivalent to a rise time of between 12.5 and 87.5% of the step response.

(viii) The LST tracker altitude rate transient errors at the time of rate change are considered to be due solely to the quantization effect. For practical purposes, the same behavior occurs every Monte Carlo pass. This is evidenced by the lack of standard deviation spread during that period.

(ix) When the altitude report reliability drops to 95% (i.e., one in twenty reports is missing, on the average) the estimation performance tends to degrade slightly. However, for practical intent and purpose, the degradation is not perceptable.

Selected Individual Cases Three selected individual cases are presented and discussed in this section. The cases presented represent performances corresponding to different altitude rate profiles of -5, 10 and 60 fps. (The 20 fps altitude rate profile was discussed in the previous section in detail.)

The extreme cases present particular challenges for the estimation task. The challenge for the 5 fps case is that the dynamic effect is lost or masked by the 100 ft quantization. New and useful information is available every 20 sec on the average. That is, the filter algorithm is "dead-reckoning" for 20 sec before the estimates can be updated. For the other extreme case of 60 fps, the challenge is that there is paradoxically too much information. The level change occurs every 1.67 sec. Therefore, small errors in the level switching time computation represents a large percentage of 1.67 sec; this induces errors in the altitude rate.

Figure 47 and 48 show the sample and statistical error time plots for the -5 fps descent profile. Initial altitude is 2,970 ft, and final altitude is 2,030 ft. These intermediate altitudes were chosen so that the Mode C reports will be oscillatory in nature between, for example, FL20 and 21 during the final level-off. The following comments and remarks are derived from the simulation results.

(i) As seen in Fig. 47, the altitude rate estimate of the αβ tracker is very noisy, whereas that of the LST tracker is much more solid. The altitude estimates show a very similar characteristic.

(ii) The initial time delay effect (~30 sec) of the LST tracker is clearly shown in the altitude and altitude rate estimates. This is caused by the basic 20 sec time delay until new and useful information becomes available. The LST tracker transient behavior at the final level-off is compounded by the intermediate altitude of 2030 ft. The Mode C reported altitudes are oscilatory, as can be seen in the altitude estimate.

(iii) The standard deviations of rate errors are 1.8 and 1.4 fps for the αβ and LST trackers, respectively. The peak mean errors are 3.9 and 4.9 fps.

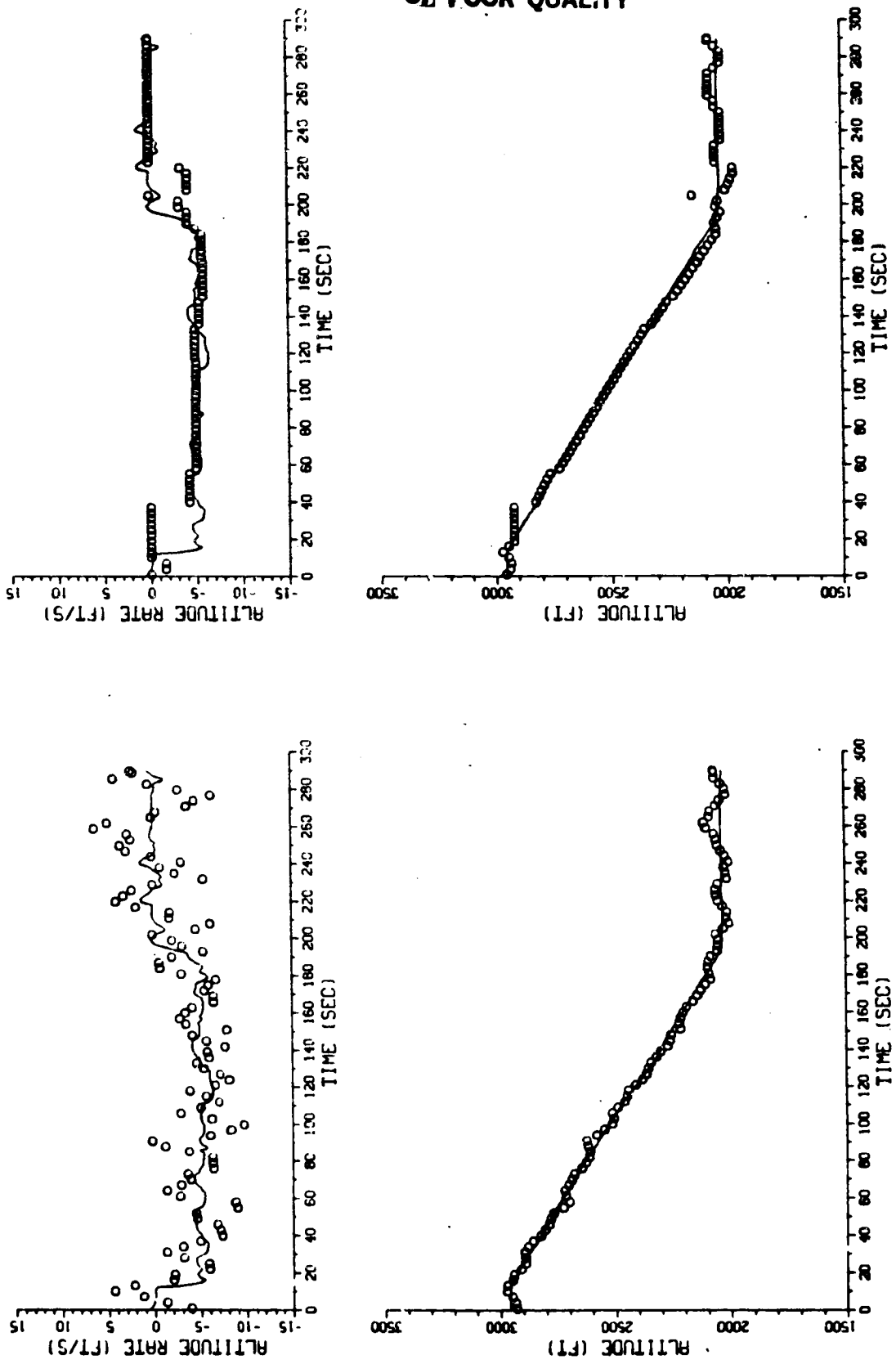
(iv) Time delay for the LST is approximately 30 sec. The transient error for the αβ tracker is masked by the high frequency error.

Figures 49 and 50 show the sample and statistical error time plots for the 10 fps climb altitude profile. Initial altitude is 2030 ft, and final altitude is 3730 ft. The following comments and remarks apply to this case.

(v) Similar to the previous case, the LST tracker's estimates are more "solid" than those of the αβ tracker.

(vi) The time delay effect (~15 sec) of the LST tracker is clearly shown in the sample time plot of the altitude rate estimate. This causes an altitude overshoot (~100 ft) at the level-off period.

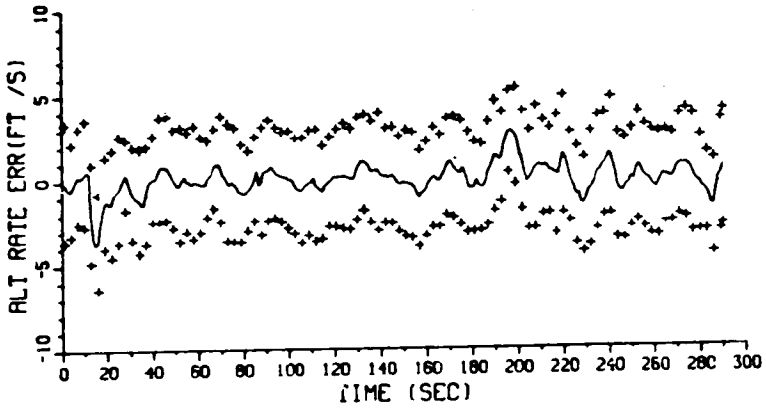
(vii) The standard deviations of the rate errors were 2.0 and 1.3 fps for the αβ and LST trackers, respectively. The peak rate errors were 4.4 and 6.3 fps. The comparable altitude errors were 27.0 and 9.8 ft for the αβ tracker and 25.8 and 52.7 ft for the LST tracker.

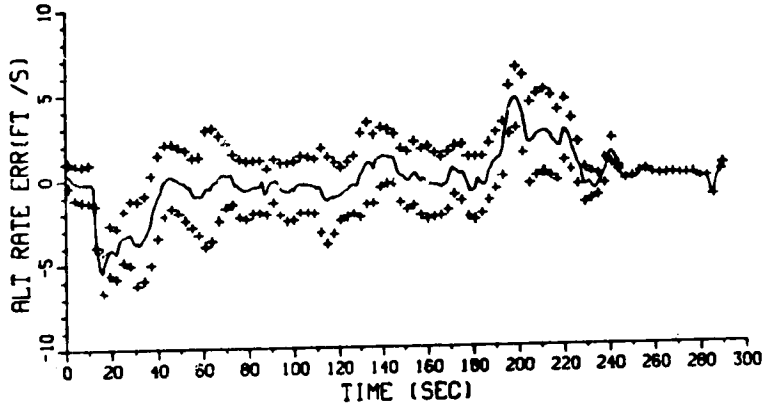


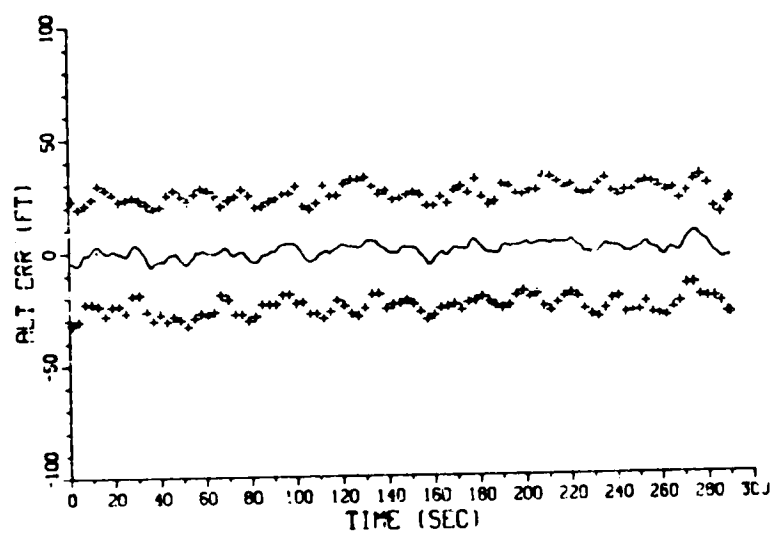
(a) αβ Tracker

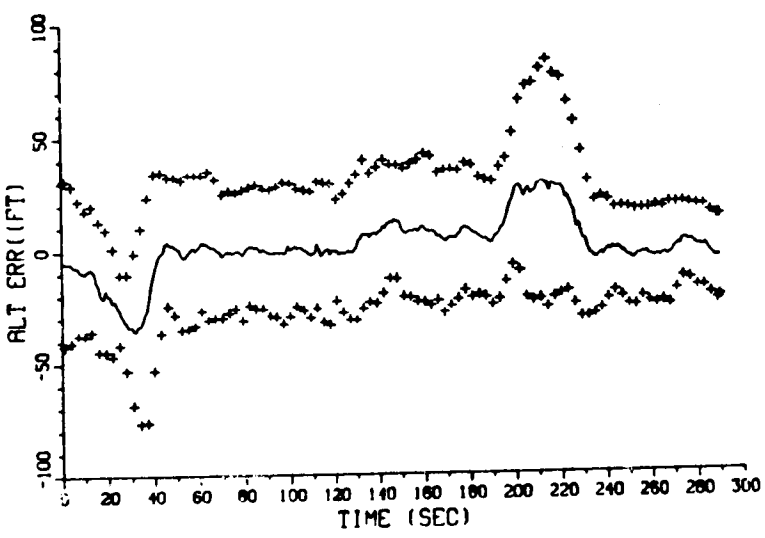
(b) LST Tracker

Figure 47. Sample Time Plots for -5 fps Profile





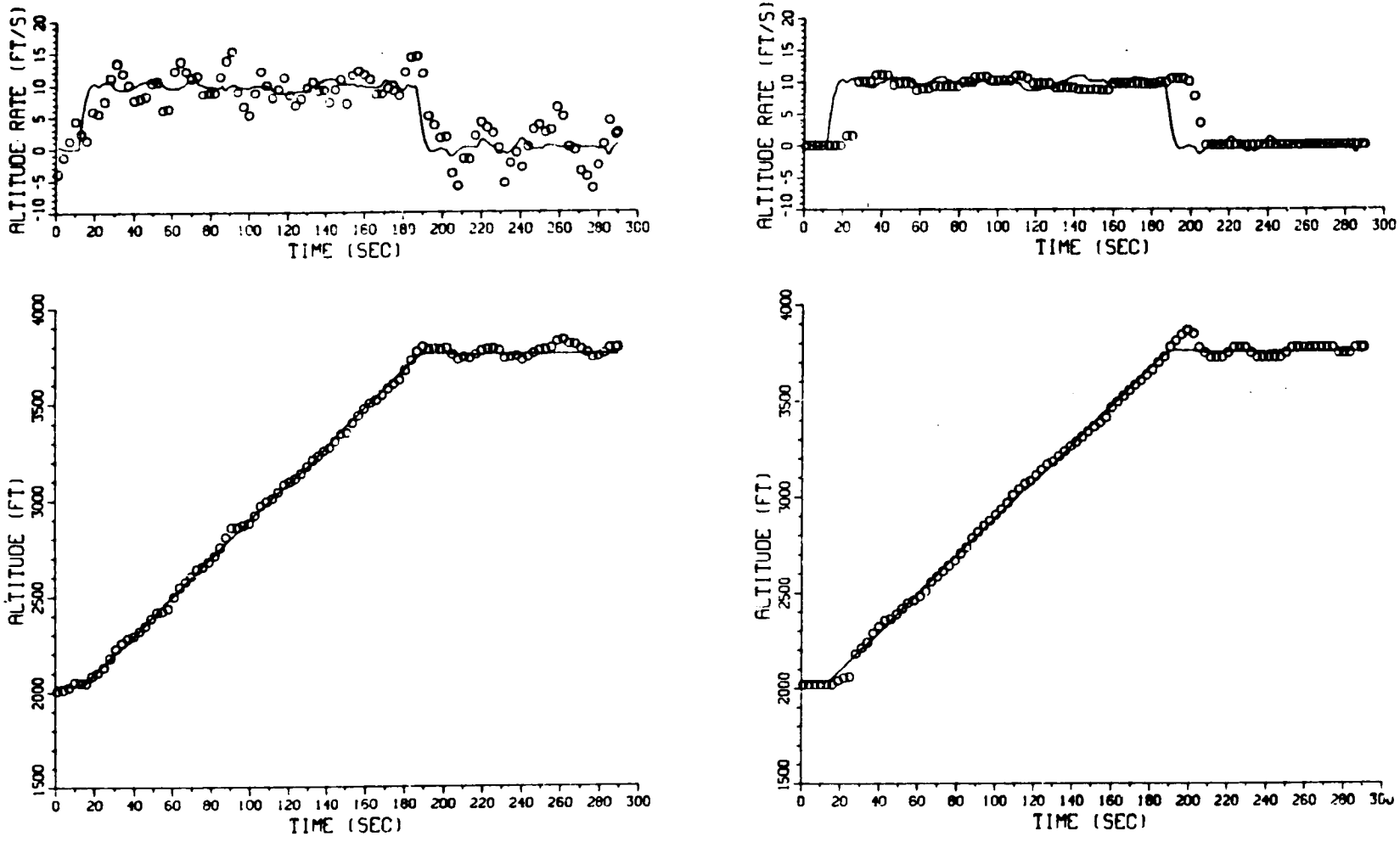




(a) αβ Tracker

(b) LST Tracker

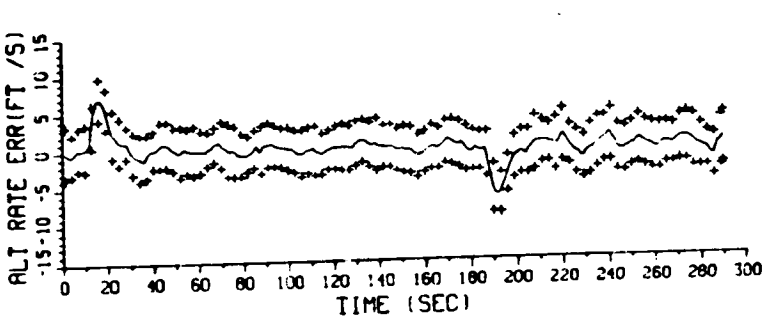
Figure 48. Statistical Error Time Plots for -5 fps Profile

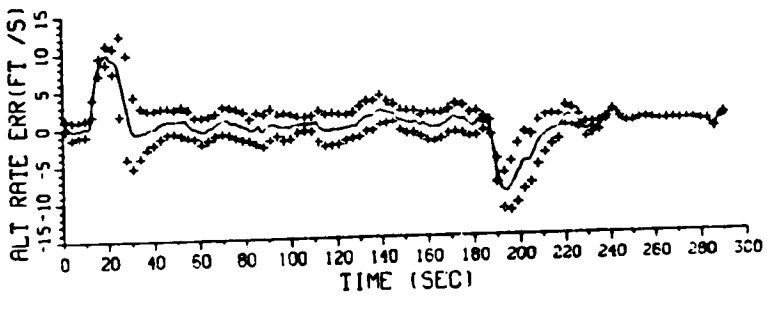


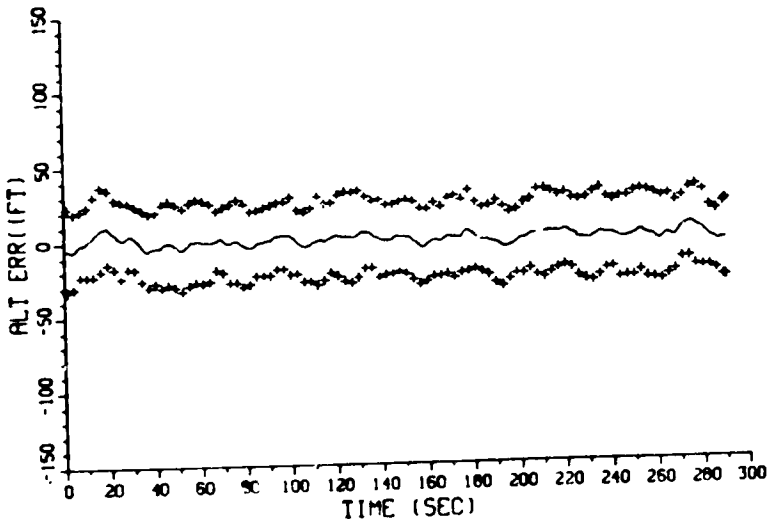
(a) αβ Tracker

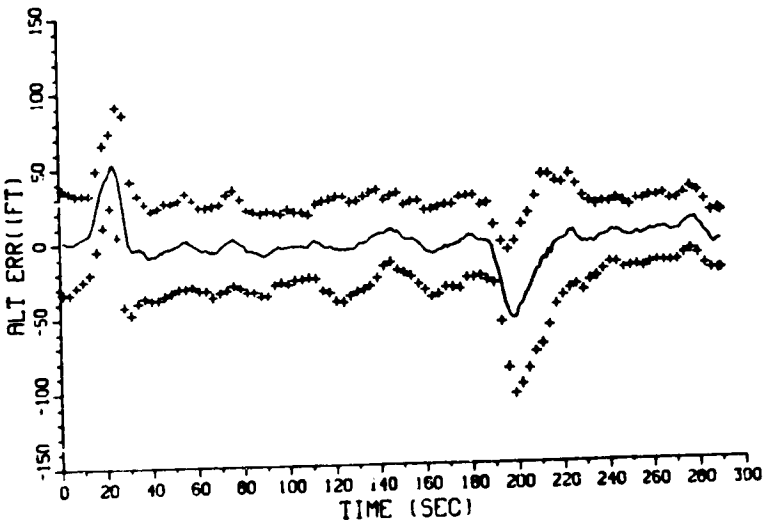
(b) LST Tracker

Figure 49. Sample Time Plots for 10 fps Profile









(a) αβ Tracker

(b) LST Tracker

Figure 50. Statistical Error Time Plots for 10 fps Profile

(viii) Dynamic time delay for the LST is approximately 15 sec, whereas that of the αβ tracker is approximately 6 sec. This applies to the transient behavior.

Figures 51 and 52 show the sample and statistical error time plots for the 60 fps climb profile. Initial and final altitudes are 1,600 and 10,360 ft. It is noted that the simple altitude select/hold logic used in the simulation overshoots the reference altitude by 350 ft. This is caused by the nonlinear elements (velocity and acceleration authority limitors) in the control loop. The following comments and remarks apply to this case.

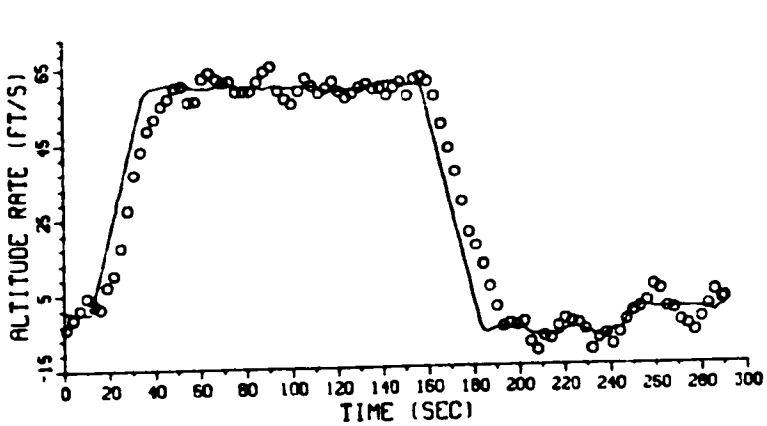
(xi) By comparing the altitude rate estimates in Fig. 51, the LST tracker estimate (in the steady state region) is noisier than that of the αβ tracker. This is caused by the deterioration in the level switching time computation accuracy.

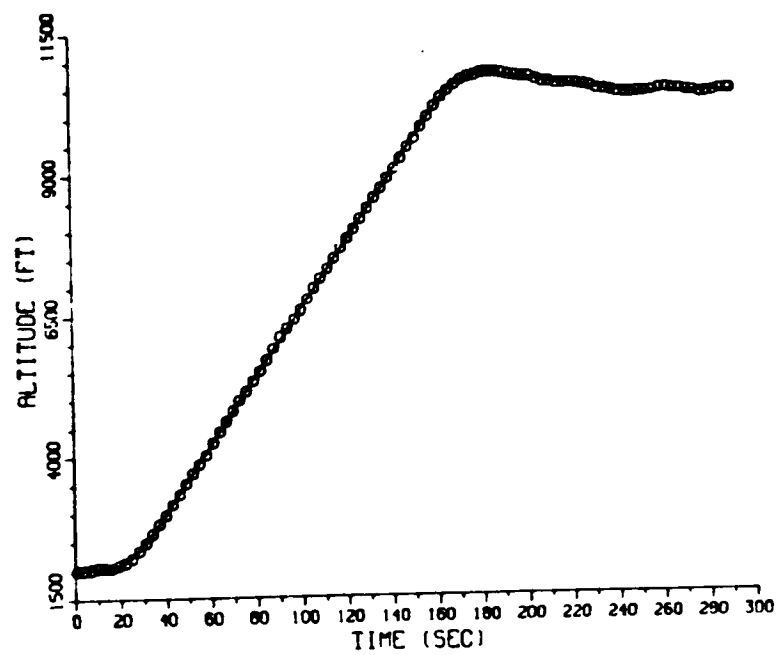
(x) The time delay effect (~6 sec) is very close to that of the αβ tracker except for the "exponential" tailing off when going level. The latter is caused by the level occupancy time becoming longer as the altitude rate becomes smaller.

(xi) The standard deviations of the rate errors were 3.1 and 4.1 fps for the αβ and LST trackers respectively. The peak rate errors were 14.7 and 21.8 fps. The comparable altitude errors where 24.8 and 17.4 ft for the αβ tracker and 29.5 and 67.1 ft for the LST tracker.

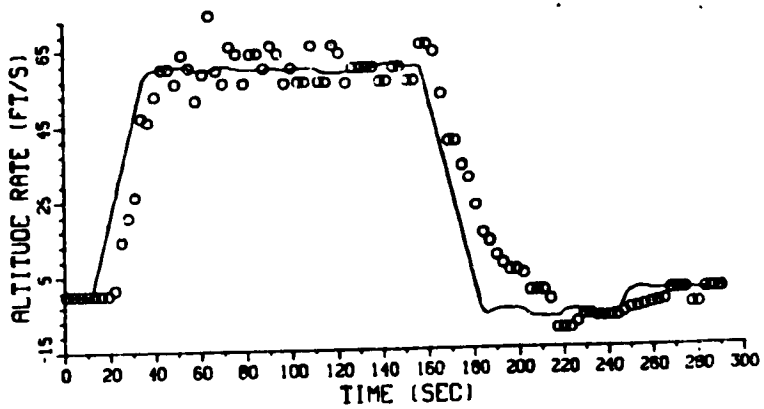
(xii) The transient period for the LST tracker is longer than that of the αβ tracker. The average periods were 26.7 and 32.4 sec. The transient period is defined to be that period where there is 6 fps and the larger rate errors.

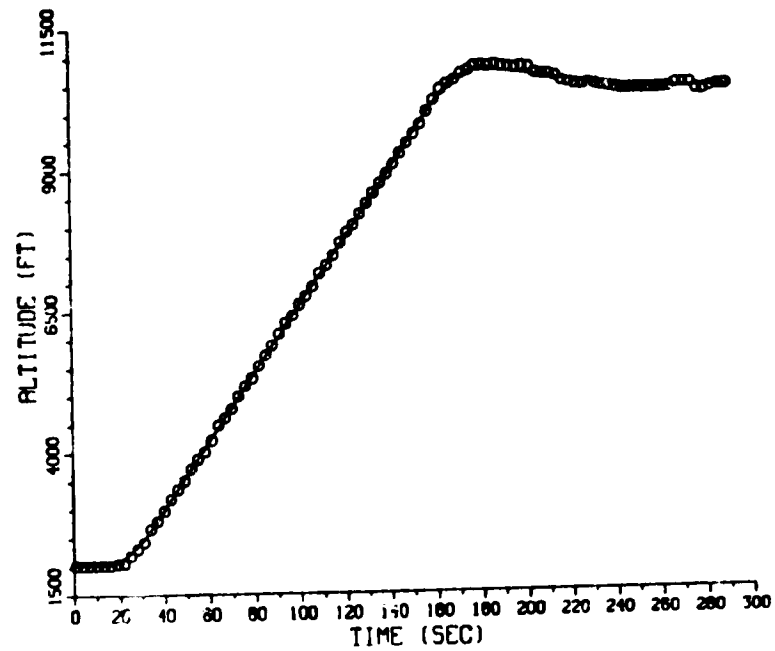
Table 11 shows the summary of the statistical errors for the four cases





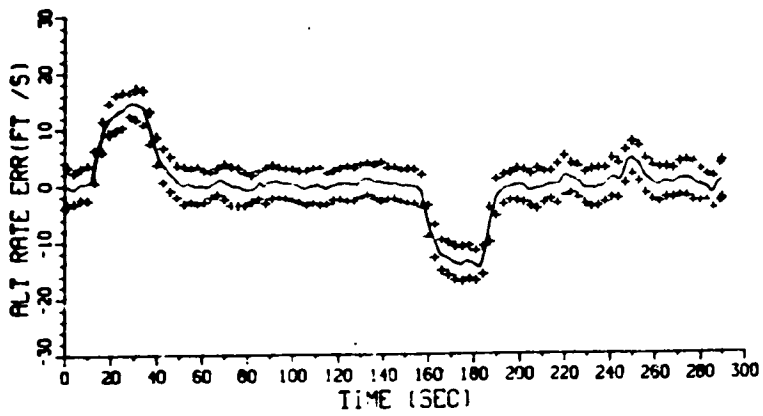
(a) αβ Tracker

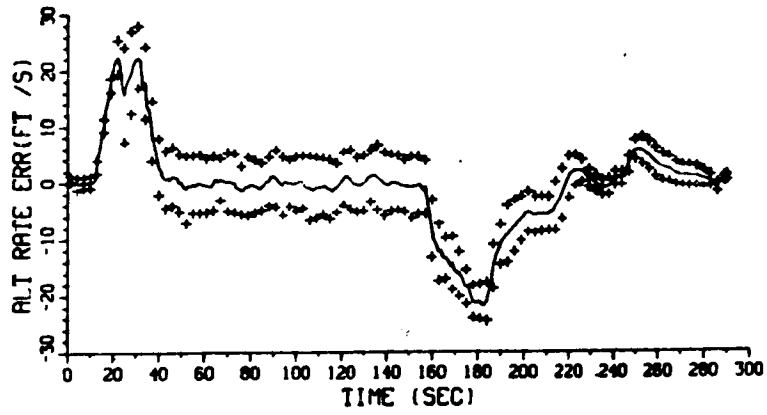


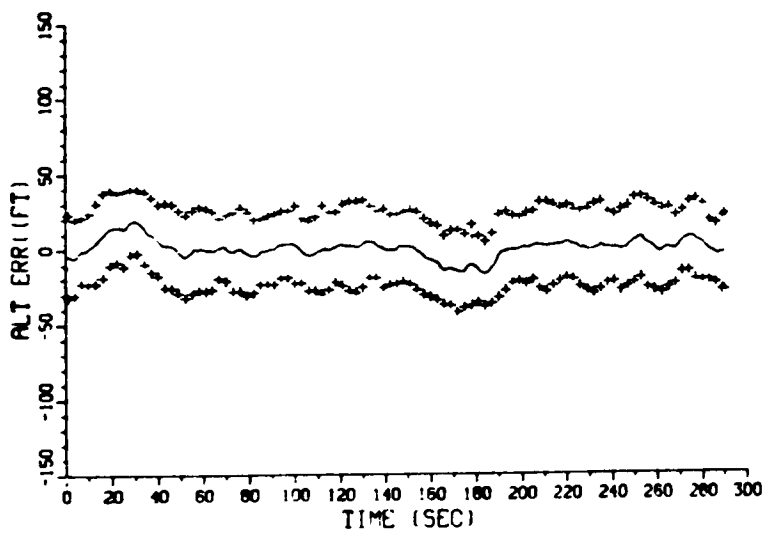


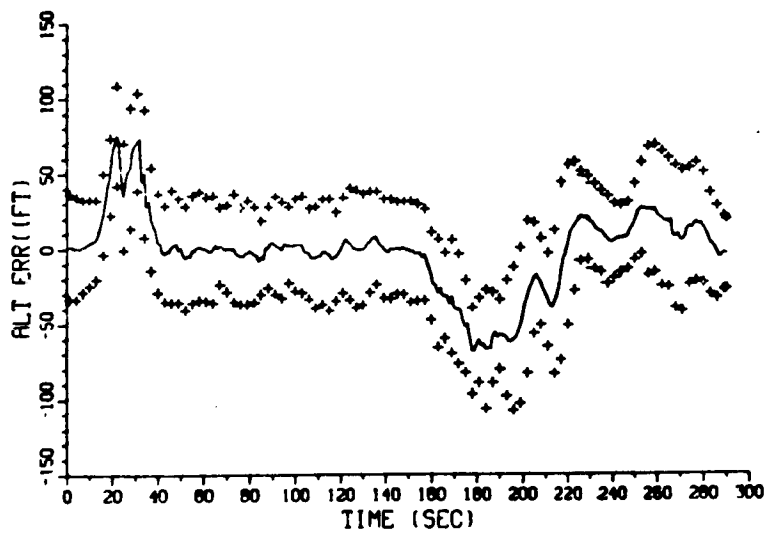
(b) LST Tracker

Figure 51. Sample Time Plots for 60 fps Profile









(a) αβ Tracker

(b) LST Tracker

Figure 52. Statistical Error Time Plots for 60 fps Profile

Table 11. Statistical Summary for Different Altitude Rate Profiles

	αβ Tracker	LST Tracker
-5 fps	22.9 (0) *	25.5 (30.7)
	1.8 (3.9)	1.4 (4.9)
10 fps	27.0 (9.8)	25.8 (52.7)
	2.0 (4.4)	1.3 (6.3)
20 fps	28.9 (23.5)	31.2 (83.9)
	3.0 (11.9)	2.4 (20.8)
60 fps	24.8 (17.4)	27.5 (69.1)
	3.1 (14.7)	4.1 (21.8)

* Altitude standard deviation (ft) Peak mean error (ft)
Altitude rate standard dev. (fps) Peak mean error (fps)

Conclusions

Based on the simulation results presented, certain conclusions can be drawn. The performance of the αβ tracker with the fine resolution altitude measurements as input is the basis for comparison.

(a) The low gain αβ tracker configuration performs better than the high gain configuration. The altitude rate estimate is smoother without sacrificing fast response time.

(b) If the additive noise is correlated (as in this study) rather than independent, the basic αβ tracker configuration needs to be modified.

(c) The LST tracker performance is very credible considering that it must work with the 100 ft Mode C quantization. In the low speed (~5 fps) and medium speed (10-20 fps) regimes the estimates are smoother than those of the αβ tracker. The peak mean errors caused by the time delay due to the 100 ft quantization are larger. In the high speed regime (~60 fps), the LST estimates are somewhat inferior.

(d) The basic dynamic delay problem with the LST tracker remains the fundamental problem. Compared to the αβ tracker the delay represents an extra ~20 sec at 5 fps and ~6 sec at 60 fps. However, this may be a limit which can not be solved by computational considerations alone. For example, it may require a cross-link of on-board generated altitude rate estimates.

In the next chapter, implications of these errors will be discussed from the collision avoidance logic point of view.

V

COLLISION AVOIDANCE LOGIC

AND

CDTI SENSOR IMPLICATIONS

In the previous three chapters, various configurations and algorithms were discussed to generate state estimates of intruder and Own aircraft relative kinematics. Performance comparisons were made in terms of "raw" estimation error statistics using Monte Carlo simulations. The comparative analyses have provided a great deal of useful information as to the relative merits. However, the information per se does not provide acceptability of a particular estimation algorithm to certain applications. The main purpose of this chapter is to discuss the estimation algorithm performance in terms of CDTI and TCAS applications.

The method of linear error analysis is mainly utilized. This method is simply stated as follows: If a dependent variable, y, is a function of several statistically independent variables, x_i, as

$$y = f(x_1, x_2, \ldots, x_n) \quad , \tag{80}$$

then the errors are related within the first order terms as

$$\delta y = \sum_i f_i(x_1, .., x_n)\delta x_i \quad , \tag{81}$$

where $f_i(\cdot)$ is the partial derivative of f with respect to x_i. The standard deviation of y can be computed by the formula

$$\sigma_y = \left[\sum_i f_i^2 (x_1, x_2, .., x_n)\sigma_{xi}^2\right]^{1/2} \quad . \tag{82}$$

Sampling Period Scheduling Logic Application

One of the unique features of the enhanced TCAS II designed by Bendix is its abiiity to time the interrogation/reply surveillance schedule under microprocessor control. This ability is enabled by a relatively narrow interrogation transmission beam-width and the beam stabilization with respect to Own aircraft orientation. The narrow interrogation beam-width, the selective interrogation time scheduling, and several levels of whisper/shout power sequencing give it the ability to operate in a very high density (up to 0.4 aircraft/nmi^2) traffic environment.

The scheduling logic criterion is given by

$$\Delta t_s = \min \{\Delta t_r, \Delta t_b, \Delta t_z, 8 \text{ sec}\} ,$$

where

Δt_r = time for the range to change 1000 ft,

Δt_b = time for the bearing to change 3 deg, and

Δt_z = time for the altitude to change 250 ft.

This is not the sole test. The other test is the target threat status. If a target is declared a preliminary threat, then the surveillance of that target is cycled at 1 sec intervals.

Range Sampling Scheduling The range scheduling time is defined mathematically by

$$\Delta t_r = |1000/\dot{r}| , \tag{83}$$

where $\dot{r}$ is the range rate expressed in fps. In reality, the schedule time is computed based on the estimates, thus

$$\Delta \hat{t}_r = |1000/\hat{\dot{r}}| . \tag{84}$$

The difference between Eqs (83) and (84) is in the precision loss induced by the range rate estimate error. If the difference is denoted by $\delta\Delta t$, the following first order equation is obtained

$$\delta\Delta t_r = -\frac{1000}{\dot{r}^2}\,\delta\dot{r} = -\Delta t_r \cdot \frac{\delta\dot{r}}{\dot{r}} \quad , \tag{85}$$

where $\delta\dot{r}$ is the estimation error in the range rate estimate. Dividing both sides by Δt_r, Eq (85) says that the positive (negative) percentage error in the range sampling schedule time is identical to the negative (positive) percentage error in range rate estimate. Based on the Range Filter Performance Summary Table 7 in Chapter III, the following Table 12 can be derived.

From Table 12 the maximum percentage deviation due to the range rate estimation error is 22%; thus, it is apparent that the range error would not significantly affect the sample scheduling time computation. One note of caution should be mentioned here. The range rate errors are computed based on the measurements available in one second intervals. Therefore, when the measurement cycle is, for example, 3.7 (or 4) sec, then the range rate error is expected to be much larger. Even if the errors were four times larger, the changes would be less than 1 sec. (The exception is that they would be 1.7 sec deviation for the range $\alpha\beta$ tracker case.)

Altitude Sampling Schedule Analogous to the range case a similar analysis can be performed for the altitude axis. The altitude error equation corresponding to the range error Eq (85) is given by

$$\delta t_z = -\frac{250}{\dot{z}^2}\,\delta\dot{z} = -\Delta t_z \cdot \frac{\delta\dot{z}}{\dot{z}} \quad , \tag{86}$$

or in terms of percentage

$$100 \cdot (\delta t_z/\Delta t_z)\ \% = -100 \cdot (\delta\dot{z}/\dot{z})\ \% \quad . \tag{87}$$

Referring to the altitude tracker statistical performance in Table 11, it is clear that the altitude sample scheduling time is not affected for the low to medium altitude rate (up to 15 fps) cases. For example, a + 100 % increase in altitude rate error represents 30 fps of estimate instead of the nominal 15 fps. Accordingly, the computed sample schedule time is 9.3 sec instead of the true 16.7 sec. In both cases, the sampling period would be 8 sec because of the imposed 8 sec minimum.

Table 12. Percent Error in Range Sample Scheduling Time Computation (sec)

Encounter		Nominal Range Rate & Sampling Time	Non-Aided	Target- & Own-Aided	Range αβ	Range Square αβγ
Tail Chase		160 kt 3.7 sec	4.3(1)	4.0	11.4	5.6
Route Crossing	1x	280 kt 2.1 sec	2.4	1.8	6.5	2.7
	2x		4.4	3.4	12.7	5.0
Head-on	1x	340 kt 1.8 sec	3.0	2.0	5.5	2.2
	2x		6.0	4.2	10.7	4.7
	4x		10.9	8.4	21.7	9.4
Parallel Turn-in	1x	0 - 400 kt 8 - 1.5 sec	7.5(2) (18.2)	3.1	10.6 (9)	7.4 (14.6)
	2x		9.9 (18.7)	5.9	19.1 (7.4)	10.6 (14.6)

(1) Percentage change to Interrogation Scheduling Time.

(2) Computed based on the relative range rate of 200 kt.

When the altitude rate is larger than 15 fps, there could be some deterioration. Over estimation of altitude rate (i.e., the magnitude of the estimate is larger than the magnitude of true state) would in fact help the estimation process by providing the measurements at a higher than nominal sampling rate. Thus, the deterioration occurs only when the altitude rate is significantly underestimated. In the 60 fps vertical profile case, the nominal sampling period is 4.2 sec. If the rate is underestimated by 15 fps as in the $\alpha\beta$ tracker case, then the computed period is 5.6 sec with the difference of + 1.4 sec. If the rate is underestimated by 22.2 fps as in the LST tracker case, the corresponding values would be 6.6 sec and + 2.4 sec. Because of the relatively long duration of the transient error with that magnitude, the longer-than-nominal sampling period would last approximately 15 sec and 20 sec for the $\alpha\beta$ and LST trackers, respectively. (It is not meaningful to discuss the $\alpha\beta$ tracker results in this application except for comparison purposes, because Own measurements are available every second.)

As mentioned previously, the above discussion is somewhat simplistic in the sense that the estimation error was generated based on the one second sampling interval and not the computed sampling rate. However, the results can be considered the expected errors under the best circumstances. It is expected that the transient errors of the LST tracker would be substantially worse at long sample intervals.

Collision Avoidance Logic Applications

The range and altitude are closely monitored by the collision avoidance system (CAS) logic in order to ascertain an intruder's threat status. Actually the CAS logic consists of two parts. One is to determine the presence of a threat (called Threat Detection Logic), and the other is to determine an escape maneuver (called Resolution Determination Logic). The main concern here is the detection performance. Basically, the (vertical CAS) detection logic consists of two parts. One is the so-called range test, and the other is the altitude test.

When the range is closing, the range test monitors the range closure time called tau (τ_r). Mathematically the test is expressed as

$$\tau_r \equiv \frac{\hat{r} - d_r}{\dot{\hat{r}}} : \theta_\tau \tag{88}$$

where d_r is the minimum range guard and θ_τ is the threshold value. When τ_r is less than the threshold, θ_τ, the intruder is said to pass the range test. The value of d_r varies from 0.075 to 1.3 nmi, and θ_τ varies from 18 to 35 sec, both depending on own altitude.

The main idea of the altitude test is to examine the projected relative altitude τ_r sec into the future. If the projection is within a threshold then the intruder is said to pass the altitude test. Mathematically, the altitude test is expressed as

$$\left|(\hat{z}_0 - \hat{z}_T) + \tau_r \cdot (\dot{\hat{z}}_0 - \dot{\hat{z}}_T)\right| : \theta_z \quad . \tag{89}$$

The altitude threshold (θ_z) value varies from 750 to 950 ft, again depending on Own altitude.

Because these tests utilize the estimates, it is necessary to examine the effect of the estimation errors on the tests. These are discussed in the following sections using linear error analysis methods. As typical parameter values, 1 nmi, 30 sec and 850 ft are used for the minimum range guard, the projection threshold and the relative altitude threshold, respectively.

Range Closure Time The linear perturbation error equation corresponding to Eq (88) is given by

$$\delta\tau_r = -\frac{1}{\dot{r}} (\delta r - \tau_r \, \delta\dot{r}) \quad . \tag{90}$$

Assuming that δr and $\delta\dot{r}$ are independent (i.e., $E\{\delta r \, \delta\dot{r}\} = 0$) the standard deviations are related by

$$\sigma_\tau = \frac{1}{\dot{r}} \left[\sigma_r^2 + \tau_r^2 \sigma_{\dot{r}}^2\right]^{1/2} \quad . \tag{91}$$

Referring to Table 7 in Chapter III and using 30 sec for τ_r, the following Table 13 is obtained. The error for the parallel turn-in case was computed using the root mean square instead of the standard deviation values at 200 kt. The following comments and remarks apply.

(i) All the tracker configurations performed well for rectilinear encounter kinematics. The worst was 6.5 sec error for the $\alpha\beta$ tracker for the head-on case when the measurement noise was four times the nominal. 6.5 sec is 22% of the nominal 30 sec. Thus, for this case, 34% of the time, the threat warning would be delayed 6.5 sec (i.e., there would be 23.5 sec left for appropriate escape maneuvers.)

(ii) The closure time errors for the parallel turn-in case vary from 4.2 to 6.3 sec for all configurations except the aided tracker. "Half" of the errors are attributable to the dynamic delay error. This implies the errors are sustained for the duration of the maneuver. Therefore, the averaging of the closure times over a few measurement intervals would not improve the error. Therefore, this would be a vulnerable period. The degradation of the $\alpha\beta$ tracker performance is less than the other two trackers. This is due to the high feedback gains.

(iii) The closure time errors are computed on a steady state basis. The errors would be larger during the initial transient period. (This is especially true for the Range square $\alpha\beta\gamma$ tracker.) This implies that the CAS protection is not reliable for pop-up targets. Pop-up targets are the ones which remain in the antenna's shadow until it is at a very close proximity of Own. In this case, the trackers would not have sufficient time to settle the initial transient errors.

In conclusion, it can be stated that all of the trackers provide good range protection for steady state rectilinear encounter kinematics. The protection becomes severely deteriorated during maneuvering periods except for the aided tracker configuration. Because of the long settling time, the Range square $\alpha\beta\gamma$ tracker does not provide good protection for very close pop-ups.

Table 13. Range Closure Time Errors (sec)

Encounter		Nominal Range Rate	Non-Aided	Target- & Own-Aided	Range $\alpha\beta$	Range Square $\alpha\beta\gamma$
Tail Chase		160 kt	1.3	1.2	3.4	1.7
Route Cross-ing	1x	280 kt	0.7	0.6	2.0	0.8
	2x		1.4	1.0	3.9	1.5
Head-on	1x	340 kt	0.9	0.6	1.6	0.7
	2x		1.8	1.3	3.2	1.4
	4x		3.3	2.5	6.5	2.8
Parallel Turn-in	1x	0 - 400 kt	5.7	0.9	4.2	4.9
	2x		6.3	1.8	6.1	5.4

Altitude Projection Error Analysis The altitude protection is provided by monitoring the projected relative altitude. The projection time is when the range is expected to be the minimum. See Eq (89). The effect of the estimation errors on the altitude protection can be analyzed by examining the linearized projection errors.

Assuming the projection period of 30 sec and utilizing the statistical errors for the altitude trackers given by Table 11, the following Table 14 of the projection errors is obtained.

The following remarks and comments are derived from the results.

Table 14. Altitude Projection Errors (τ=30 sec)

Vertical Rate	αβ Tracker	LST Tracker
5 fps	76.9(1) (117.0)(2) 140.0(3)	67.5 (177.7) 190.1
10 fps	87.0 (141.8) 166.4	64.8 (241.7) 250.2
20 fps	118.9 (380.5) 398.6	103.2 (707.9) 715.4
60 fps	117.8 (458.4) 473.3	152.5 (723.1) 739.0

(1) standard deviation
(2) worst mean error
(3) root mean square of (1) and (2)

(i) The range of errors for the αβ tracker is

standard deviation: 76.9 ~ 118.9 ft;
peak mean error : 117.0 ~ 458.4 ft;
rms of these two : 140.0 ~ 473.3 ft.

These errors apply to projecting Own altitude 30 sec into the future.

(ii) The range of errors for the LST tracker is

standard deviation: 64.8 ~ 152.5 ft;
peak mean error : 177.7 ~ 723.1 ft;
rms of these two : 190.1 ~ 739.0 ft.

These errors apply to projecting the intruder altitude 30 sec into the future.

(iii) If the above sets are combined in an rms sense, then the relative projection errors are obtained.

standard deviation:		100.6 ~ 193.4 ft;
peak mean error	:	212.8 ~ 856.2 ft;
rms of these two	:	235.4 ~ 877.7 ft.

(iv) The worst combined errors represent 27.7 to 103.3% of the altitude separation threshold of 850 ft.

(v) The major contributor of the combined errors is the peak mean errors of the LST tracker. The source of this error is, of course, the inherent tracker error introduced by the 100 ft quantization. Thus, during this transient period, the LST estimates would not provide sufficient protection.

(vi) During the constant altitude rate flight (for both intruder and Own), the combined errors are less than 200 ft (23.5% of 850 ft). If no other error source is present, the estimation precision may be sufficient to gauge the threat situation.

(vii) The estimation error statistical data were obtained by assuming a high frequency error of $\pm$ 23.8 ft $\pm$ 1σ and the Mode C 100 ft quantization. Low frequency errors such as bias, scale factor or pressure transducer dynamic delay errors were not considered. For complete assessment, these error sources need to be factored into the analysis.

Three conclusions can be drawn based on the above simple analysis.

(a) If both intruder and Own maintain steady altitude rates, then the vertical threat assessment can be made with sufficient precision.

(b) During transient periods, the dynamic delay errors may not allow accurate threat assessment.

(c) If the combined low frequency (bias, scale factor, or drift) error is 200 ft or larger, then the vertical threat assessment accuracy becomes marginal.

One vs. Two Dimensional Horizontal Test The range test given by Eq (88) is called the modified tau test. The more standard test is given by

$$\tau_r \equiv -\frac{r}{\dot{r}} : \theta_r \quad . \tag{92}$$

τ_r can be interpreted to be the time to collision ($r + \tau_r \dot{r}=0$). When the relative kinematics are non-accelerating and when the aircraft are in collision courses, the above interpretation is correct. However, it is no longer correct if the miss-distance is non-zero. In fact, the test Eq (92) is inefficient in the sense that it passes many non-threatening intruders; these are unnecessary alarms.

The one dimensional range test Eq (92) is designed for range-altitude TCAS, especially the so-called active BCAS which is the direct predecessor to the current minimum TCAS II. It does not require a directional (bearing) capability. In light of the added enhanced TCAS II capabilities, a better two dimensional test is available utilizing accurate bearing measurements as well as stabilization with respect to Own attitude orientation.

Now assuming rectilinear motions, range is given by

$$r(t) = [(x + t\dot{x})^2 + (y + t\dot{y})^2]^{1/2} \quad , \tag{93}$$

where x, y, $\dot{x}$ and $\dot{y}$ (Δ notation is dropped here) are constant. The so-called time to closest point of approach (τ_{CPA}) is defined to be the value of t which minimize r(t). The miss distance (m_d) is the minimum range. These two quantities are computed by the following formula:

$$\tau_{CPA} = (x\dot{x} + y\dot{y})/v^2 \quad ; \quad m_d = |x\dot{y} - y\dot{x}|/v \tag{94}$$

where v is the relative speed defined by

$$v = [\dot{x}^2 + \dot{y}^2]^{1/2} \quad .$$

It is noted that τ_{CPA} and τ_r are in general different.

The two dimensional horizontal test would take the following form

$$\tau_{CPA} \leq \theta_{\tau} \quad .\text{AND.} \quad m_d \leq \theta_m \quad , \tag{95}$$

where θ_{τ} is the same as in Eq (88) and (92), and θ_m is the minimum miss-distance threshold. (θ_m may or may not be the same as d_r.) Actually, the condition "$m_d \leq \theta_m$" is sufficient to identify threat status as long as τ_{CPA} is positive. The condition "$\tau_{CPA} \leq \theta_{\tau}$" states that one can wait until θ_{τ} sec to go before he must take an evasive action.

The range can be expressed in terms of τ_{CPA} and m_d as

$$r(t) = [m_d^2 + v^2(t - \tau_{CPA})^2]^{1/2} \quad , \tag{96}$$

by defining time-to-go, $t_G = \tau_{CPA} - t$, then range-to-go becomes

$$r(t_G) = [m_d^2 + v^2 t_G^2]^{1/2} \quad . \tag{97}$$

The range closure time, τ_r, is given by

$$\tau_r = \frac{m_d^2 + v^2 t_G^2}{v^2 t_G} = \frac{(m_d/v)^2 + t_G^2}{t_G} \quad . \tag{98}$$

Figure 53 shows the relationship between τ_r and t_G. When the miss distance is zero (i.e., collision), then the range closure time is the same as time-to-go.

When an intruder is identified to be a near-miss by the two dimensional test Eq (95), then it automatically satisfies the modified range tau test Eq (90), if the intruder is closing at all. However, the standard range tau test (92) may or may not be satisfied. In the case of a near-miss with m_d = 0.3 nmi and θ_c = 30 sec, the test is satisfied if the relative speed, v, is equal to or greater than 72 kt; otherwise, it would not be.

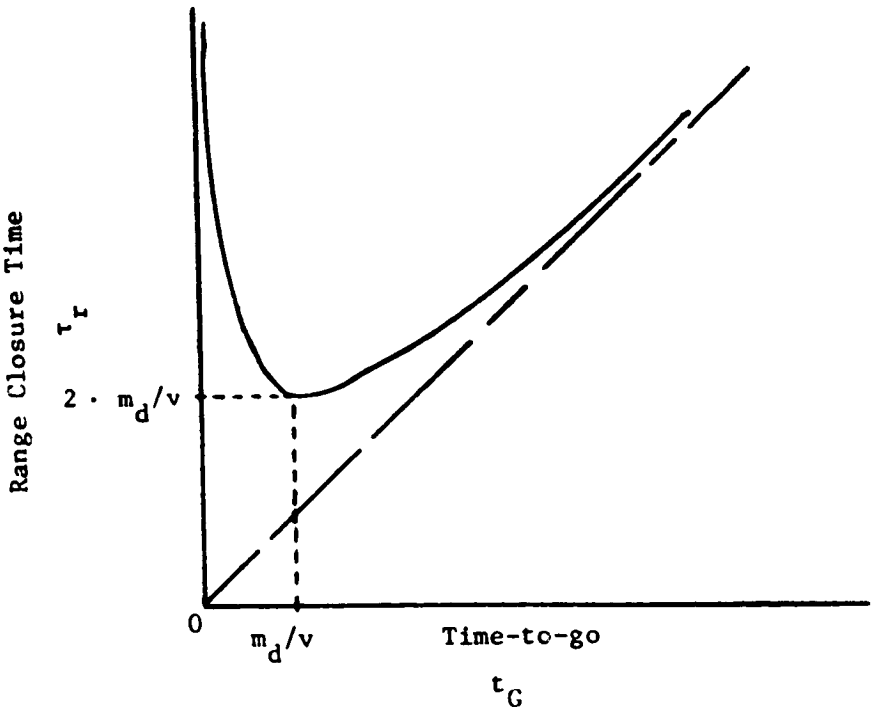


Figure 53. Range Closure Time as a Function of Time-to-go

There are numerous (infinite) unnecessary alarm examples of passing the standard or modified range test but not the two dimensional test. A typical example is shown in Fig. 54. At t=0, Own flies due north at 200 kt. Intruder flies with 150 deg heading at 300 kt. The initial intruder position is exactly 3.5 nmi due north of Own. The following CAS parameters can be computed:

miss distance, m_d	=	1.08 nmi
time to CPA, τ_{CPA}	=	24.75 sec
range @ t=0	=	3.5 nmi
range rate @ t=0	=	-459.8 kt
standard tau @ t=0	=	27.38 sec
modified tau @ t=0	=	19.56 sec.

If the miss distance threshold is 1 nmi, the intruder will not be classified as a threat, since m_d > 1 nmi. τ_{CPA} states the miss distance is reached in 25 sec. However, if the tau threshold is 30 sec, then both

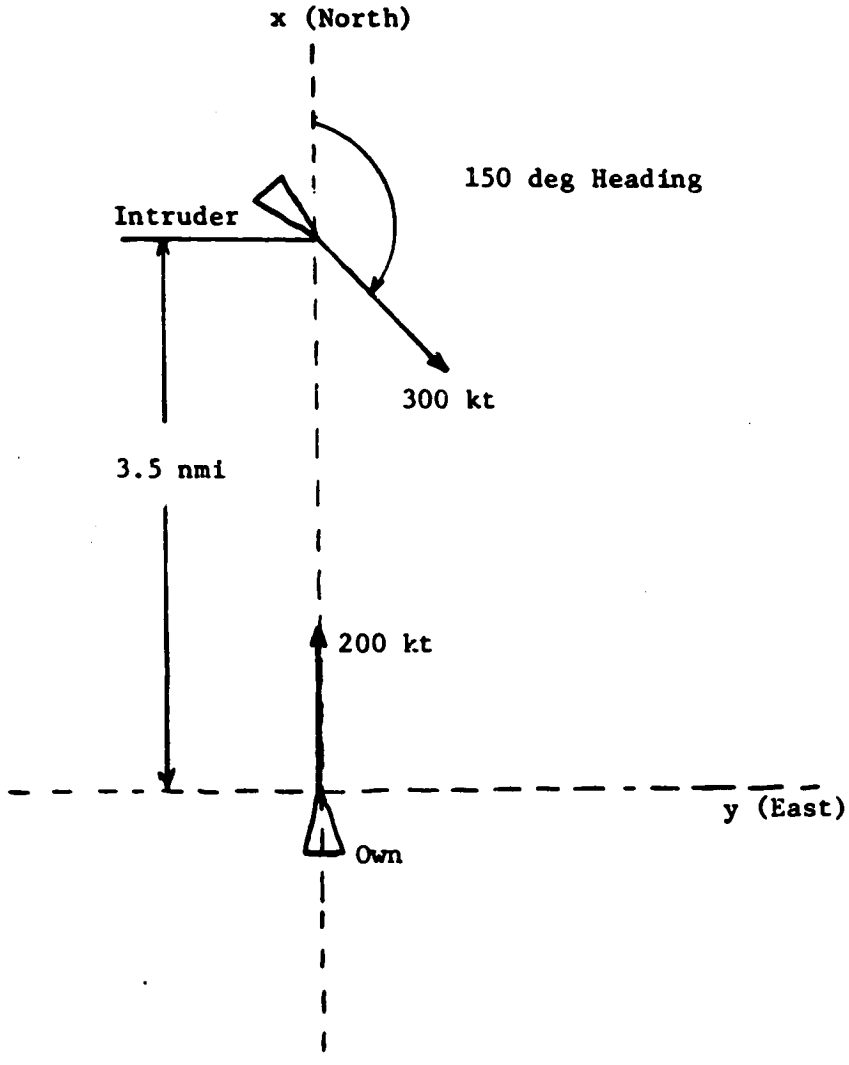


Figure 54. Relative CAS Geometry for the Counter Example

the standard and modified range tau tests are passed, i.e., this intruder will be classified as a threat. Therefore, this is an instance of unnecessary alarm. The above discussion and example show that the two dimensional horizontal test is a stronger and a better threat detection criterion than the one dimensional range tau tests.

Error Analysis The next step is to determine how well these parameters can be determined from the state estimates. The linear error analyses of these parameters with respect to the estimation errors becomes very complex. Therefore, the Monte Carlo simulation was performed. The simulation scenario was the head-on encounter (Case (C) of Fig. 14). True miss-distance is 0.31 nmi.

The Monte Carlo simulation program was modified to compute the miss-distance (m_d) and the time to the Closest Point of Approach (τ_{CPA}). These are computed based on the true state variables, the non-aided x-y filter state estimates and the Target- and Own-Aided x-y Kalman filter state estimates. The measurement update period is 1 sec. Simulations were run for the nominal, twice the nominal and four times the nominal range and bearing errors. To repeat, the nominal is defined to have a ranging error of $\pm$ 75 ft ($\pm 1\sigma$) and a bearing error of $\pm$ 1 deg ($\pm 1\sigma$). The errors are assumed to be independent white noise.

Table 15 shows the summary of simulation results. It is organized to show the error dependence on the range as well as the noise level. For example, at range of 5.7 nmi, the time to the closest point of approach is 60 sec. At this point in this particular encounter, the m_d and t_{CPA} errors based on the Non-Aided filter estimates with the nominal measurement errors were 0.35 nmi and 1.87 sec, respectively. This means that 68% of the time the m_d estimate is between 0.0 to 0.66 nmi compared to the true value of the 0.31 nmi. If the miss-distance protection is 1 nmi, then 95% (2σ band) of the time the test would make a correct assessment. The following comments and remarks are derived from the simulation results.

Table 15. Summary of Miss-Distance and Time to CPA Errors

Range τ_{CPA}	Non-Aided Filter		Aided Filter		Measurement Error Level
	m_d (nmi)	τ_{CPA} (sec)	m_d (nmi)	τ_{CPA} (sec)	
5.7 nmi 60 sec	0.35(1)	1.87(2)	0.24	1.13	nominal
	0.73	4.17	0.43	2.40	2 times
	1.48	9.53	0.90	5.33	4 times
4.2 nmi 45 sec	0.27	1.13	0.19	0.67	nominal
	0.51	2.67	0.31	1.43	2 times
	1.03	6.6	0.64	3.53	4 times
2.7 nmi 30 sec	0.16	0.73	0.11	0.43	nominal
	0.28	1.53	0.20	0.90	2 times
	0.52	3.47	0.34	2.03	4 times

(1) miss-distance standard deviation (nmi)

(2) time to CPA standard deviation (sec)

(i) The errors are, for all practical purposes, proportional to the TCAS sensor error magnitude.

(ii) At the range of 4.2 nmi (45 sec to CPA), the non-aided tracker would make a 100% correct assessment, if the measurement errors are nominal. This drops to 68% at twice the nominal. At four times the nominal values, the ratio would be less than 50%.

At the same range, the aided filter would make a correct threat assessment of 100% up to two times the nominal. At four times level, the reliability drops to 68%.

The τ_{CPA} errors are not significant, the maximum being 6.6 sec. This would still afford sufficient protection time.

(iv) At the range of 2.7 nmi (30 sec to CPA), the estimates improve substantially. Both trackers would make a correct assessment of almost 100% of the time with up to two times the nominal error.

At four times the nominal, the reliability drops to 65% and 90% for the non-aided and aided filter respectively.

At this range, the worst τ_{CPA} error was 3.5 sec for the non-aided tracker at four times the nominal measurement error level. The error represents 11%. This magnitude of time error is thought to be noncritical.

It should be strongly emphasized that the above comments are based on a single simulation. The reliability numbers are based on the assumption that the errors are normally distributed. The analysis should not be taken at the 10^{-6} type precision. With these caveats, an important conclusion emerges; at the critical time of 30 sec-to-go, the two dimensional tests can assess threat status fairly accurately. One corollary is that the threat assessment becomes more accurate as the range becomes closer.

CDTI Applications

The CDTI's major function is to provide the surrounding traffic information base to the pilots. Therefore, the CDTI estimation accuracy requirements depend on how the pilots choose to use it. Obviously very accurate position information is not needed if the pilots want to be simply aware of proximite traffic. If the CDTI is used to perform sophisticated Electronic Flight Rule (EFR) tasks, then highly accurate estimates may be needed. The EFR tasks may include self-spacing along a route, route-crossing, merge into a traffic stream and so on.

Because of the human factor element in the CDTI applications the only meaningful way to determine the CDTI accuracy requirements is through pilot-in-the-loop simulation study. However, in this environment, other issues come into play such as display size, brightness, contrast, symbology and so forth. A recent simulation study [20] obtained two results concerning parameters relating to CDTI estimation accuracy requirements.

(a) Displayed traffic position errors with standard-deviation values up to 0.3 nmi range and 8° azimuth had negligible effect on the ability of the pilots to perform the self-spacing task; and

(b) Display of the lead aircraft groundspeed was found to affect the mean spacing performance, especially during periods of speed or spacing changes. Pilot comments cited the ground-speed information as a definite aid in performing the spacing task.

These must be considered as two separate factors, since it is not possible to obtain accurate ground speed estimates when the measurement error magnitudes are 0.2 nmi and 8 deg for the range and bearing, respectively. It is also interesting to note that the ground speed information was used by the experiment pilots as a damping signal to prevent spacing "overshoot".

It is safe to say that the accuracy per se will not be a major issue unless, for example, an in-trail following "flight director" signal was generated and incorporated as an integral part of the CDTI symbology. In such a case, the connections between the (pilot) performance, the flight director accuracy,and the underlying state estimation accuracy can be

analyzed in a more qualitative fashion rather than through a more quantitative simulation analysis. It is very difficult to specify estimation accuracy requirements for the CDTI applications in a comprehensive quantitative way.

In the following sections, effects of measurement and corresponding estimation errors are discussed with respect to selected CDTI variables. Also discussed are smoothing (as opposed to filtering) algorithms for selected CDTI applications.

Bearing Errors In many cases, the surrounding traffic information is superimposed with other symbology on an EHSI (Electronic Horizontal Situation Indicator) or an MFD (MultiFunction Display). Other information may include reference air-route, map and terrain information, waypoints and navaids. Thus, the traffic position would be referenced to a local-level map fixed coordinate system. (See Fig. 55 .) This can be accomplished by transforming the TCAS measurements - relative range (r), bearing (b) and altitude (z) - to a north referenced Own fixed local level coordinate system utilizing Own body attitude angles. To this relative NED position is added Own earth-fixed NED position to obtain the "true" horizontal projection. This process is explained in Appendix A. (See Fig. A-1.) When Own attitude angles are not properly accounted for, then the horizontal projection could be substantially in error [21]. The angle, $b+\psi$ (TCAS relative bearing plus Own heading), is not the true horizontal bearing with respect to north, when the Own roll or pitch angles are non-zero. For a 20 deg roll angle, for example, the peak error (depends on (ψ) would be 5 deg if target elevation is 10 deg (Fig. 56). Thus, it is safe to say that proper transformation must be performed either within or outside the TCAS processor for the CDTI application.

Assuming that the above problem is solved, we need to discuss high frequency error magnitude. The draft TCAS MOPS specifies the error magnitude as 9 deg rms. Flight test results of one pre-production TCAS, designed by Dalmo Victor, showed the bearing error of 5-10 deg rms [22]. Bench test results of TCAS Engineering Unit designed by Bendix showed the error magnitude between 0.6-2 deg (1σ) [23]. Some of these numbers are preliminary.

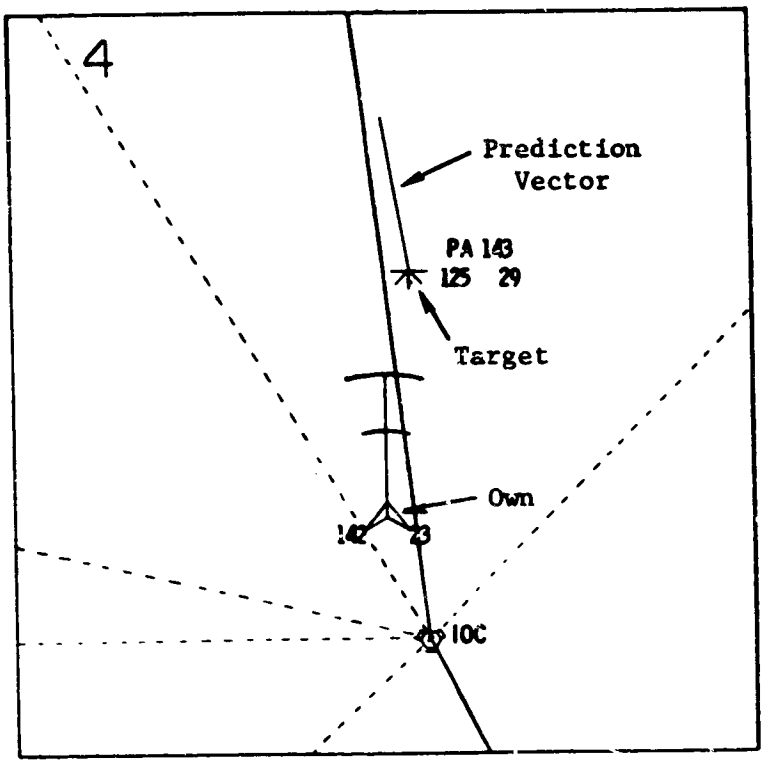


Figure 55. Example CDTI Format

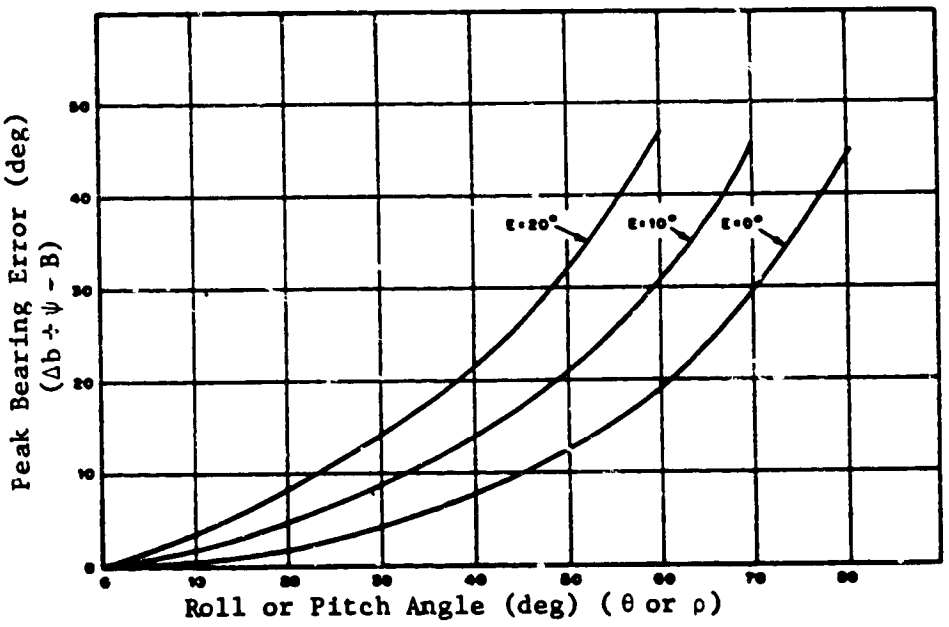


Figure 56. Peak Uncompensated Bearing Error

Considering the findings of the aforementioned NASA study, these error magnitudes are within the maximum allowable for the CDTI applications, provided that some filtering is performed prior to displaying it to the pilots. Certainly, these are sufficiently accurate in terms of clock angle indications to aid pilots in VFR applications.

According to the TCAS MOPS, the following Cartesian αβ estimation algorithm has been used successfully to develop smoothed target bearing estimates from reply measurements.

Initiation The bearing and range measurement of each reply are used to form x and y position: $x = r \cos(b)$ and $y = r \sin(b)$. The first three measurements are used to form a least-squares estimate of the x and y positions and velocities.

Prediction A predicted x,y position for the next scan is formed by adding the product of the last-scan velocity estimate and the time since the last scan to the last-scan position estimate.

Update If a valid measurement is available it is combined with the range measurement to form the x and y measurements. The update is made using a standard αβ tracker algorithm in both x and y. The gains are the same in the x and y coordinates. The gains vary during the first eight scans following initiation, and then are held constant, as shown below.

Track age	α	β
4 sec	0.700	0.300
5	0.600	0.200
6	0.524	0.143
7	0.464	0.107
8	0.416	0.083
9	0.378	0.067
10	0.345	0.054
11 or more	0.318	0.045

The position update equation in x is:

$$x(t)\ \text{estimate} = x(t)\ \text{prediction} + [\alpha * (x(t)\ \text{measurement} - x(t)\ \text{prediction})] .$$

The velocity update equation in x is:

$$\dot{x}(t)\ \text{estimate} = \dot{x}(t-T)\ \text{estimate} + [\frac{\beta}{T} * (x(t)\ \text{measurement} - x(t)\ \text{prediction})] ,$$

where T is the time difference between the current and previous measurements.

The y equations are analogous.

Bearing Computation The updated x and y are converted to bearing using the tangent function.

It should be commented that the above filtering algorithm is used to correlate transponder replies and target tracks internally stored in the TCAS surveillance processor. The algorithm would not provide good position (or bearing) estimates for the CDTI applications if Own aircraft undergoes a maneuver (either pitch or roll). This point was explained previously in this section. A better way to obtain required estimates for the CDTI applications is (1) to obtain the north referenced local level x and y measurements using Own, ϕ, θ and ψ and r and b from the TCAS, then (2) to use the $\alpha\beta$ tracker algorithm described in Chapter II to obtain the horizontal state estimates from which the bearing estimate can be computed. As mentioned previously, the attitude stabilization can be performed in the CDTI processor module outside the TCAS as long as timing is synchronized.

Target Prediction Vector Error The target prediction vector is sometimes included in the CDTI symbology - see Fig. 55. The Own pilot can extract valuable target short term future information from this vector.

The prediction vector is computed based on target ground speed (v_G) and ground track angle (ψ_G). These variables are given by

$$\hat{v}_G = [(\hat{x}_0 + \Delta\hat{x})^2 + (\hat{y}_0 + \Delta\hat{y})^2]^{1/2} , \quad \text{and} \tag{99}$$

$$\hat{\psi}_G = \tan^{-1} [(\hat{y}_0 + \Delta\hat{y})/(\hat{x}_0 + \Delta\hat{x})] , \tag{100}$$

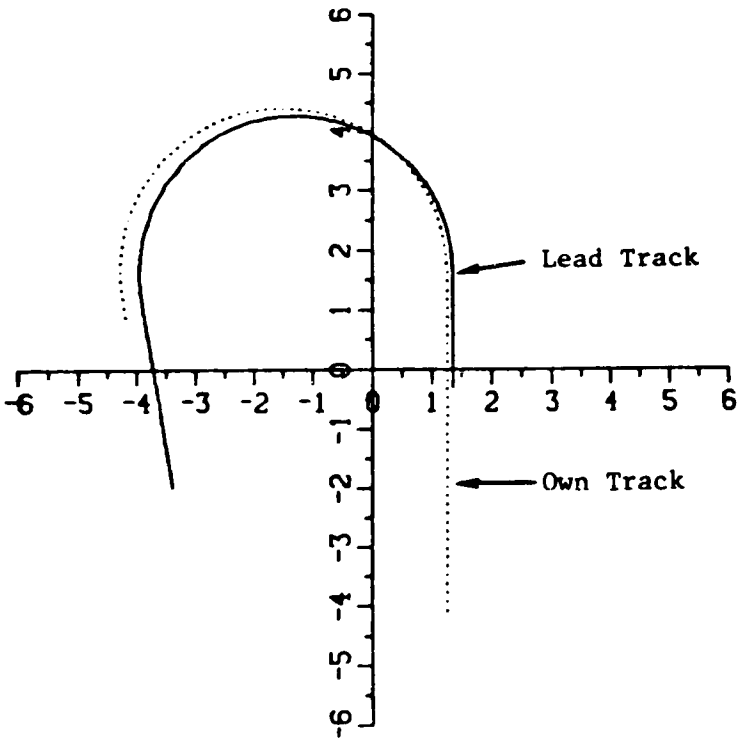
where $\hat{x}_0$ and $\hat{y}_0$ are Own horizontal velocity components provided by the on-board navigation system, and $\Delta\hat{x}$ and $\Delta\hat{y}$ are the TCAS relative velocity estimates. Obviously, the ground speed and ground track errors would depend not only on the TCAS errors but also on the on-board navigation system.

The navigation errors vary according to available navaids, configuration and geometry, body rate sensor or INS, and so on. The target ground speed is sometimes included in the target data tag. Numerical values of the ground speed and flight level are shown along with the target identification. Thus, the Own pilot would have a ready reference with respect to his Own instrument reading.

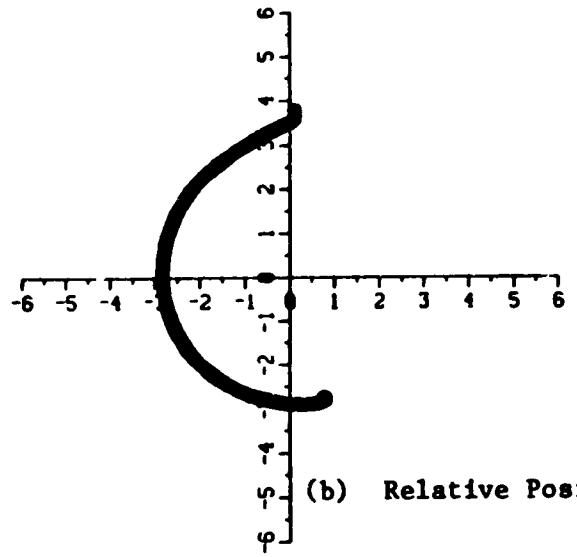
In order to assess the TCAS sensor effect on the above parameters, a simple in-trail following scenario was incorporated into the Monte Carlo simulation program. The ground speed and track angle estimates were computed according to Eqs (99) and (100), except the Own velocity components were used. The error statistics between the estimated and true variables were obtained from this simulation.

The in-trail following scenario was a simple one simulating downwind turn to final. The lead aircraft was placed 3.8 nmi ahead of Own flying due north at 200 kt. After flying straight for 1.8 nmi, the lead executes a 170°, 15° bank angle left turn, and then flies due south. Own follows the lead by flying due north at 220 kt and executing a similar turn at approximately the same location. These were performed open-loop, and the initial conditions and flight parameters were made slightly different so that the lead and Own would not traverse the same trajectory. Figure 57 shows the horizontal projections.

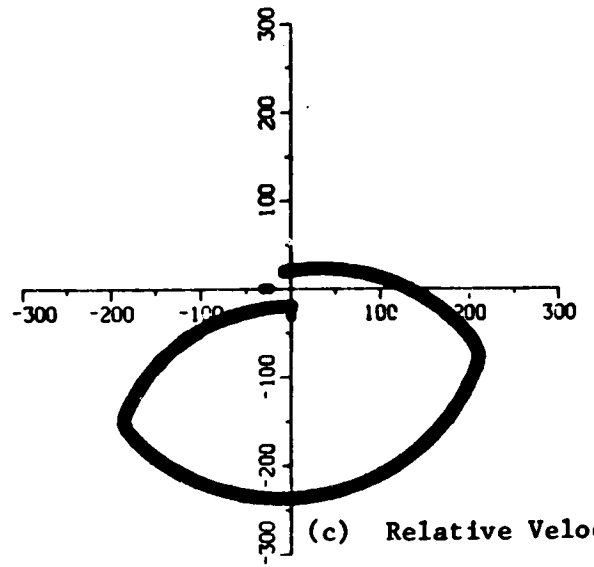
Figure 58 shows statistical error time plots of the ground speed and track angle estimates based on the non-aided x y tracker configuration. Only the non-aided tracker results are presented and discussed. For the aided tracker configuration, the estimation accuracy of the target ground speed and tracker angle does not directly reflect the CDTI position sensor accuracy.



(a) Horizontal Projection

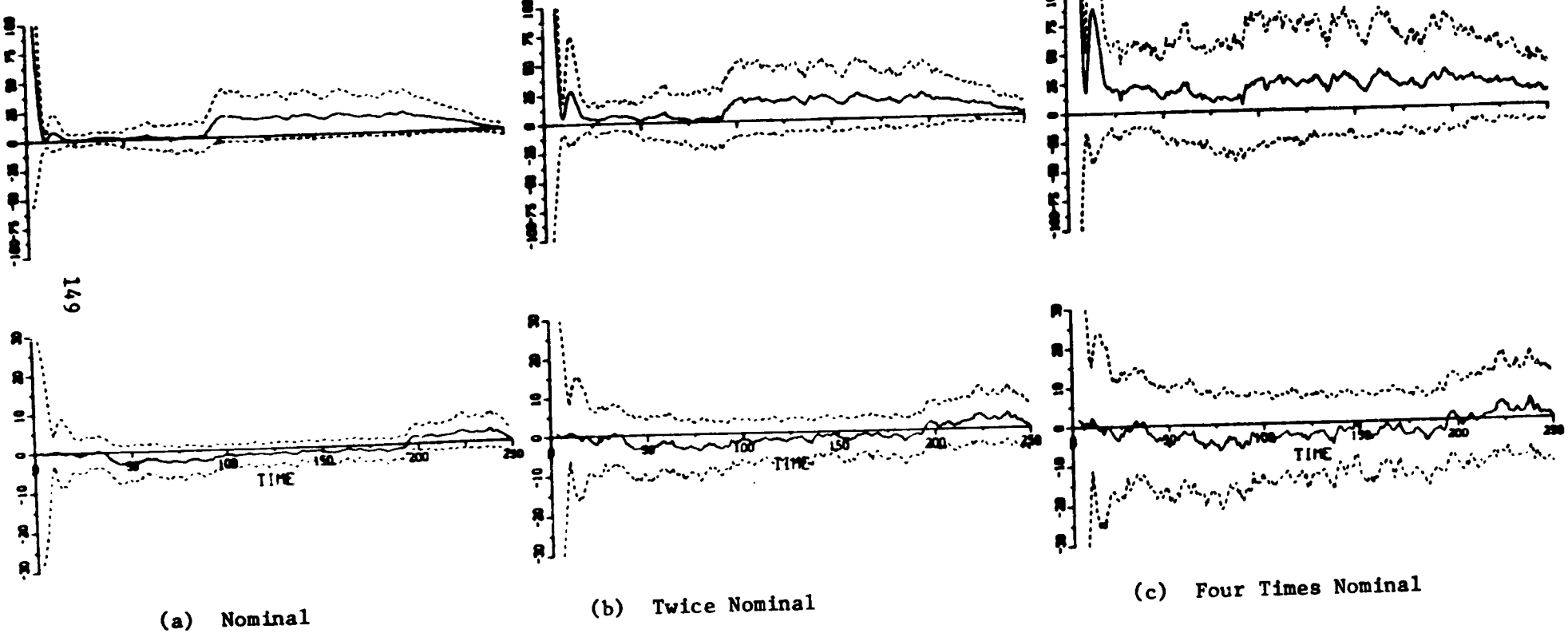


(b) Relative Position



(c) Relative Velocity

Figure 57. CDTI In-Trail Following Scenario



(a) Nominal

(b) Twice Nominal

(c) Four Times Nominal

Figure 58. Ground Speed and Heading Angle Statistical Error Time Plots for the CDTI In-Trail Scenario

The simulation data were taken for three levels of sensor measurement error: nominal, twice, and four times the nominal. The following comments and remarks are derived from the results.

(i) As expected, the high frequency errors are dependent on the relative position and velocity. When the lead is directly ahead of Own and aligned, the ground-speed error is proportional to range error, but the heading error is more affected by the bearing error. However, when the alignment becomes more oblique, the range and bearing errors are no longer orthogonal; thus, mixing into both speed and angle estimates.

(ii) This can be seen in all three ground speed error plots. After the initial transient period, the range standard deviation is small; as the bearing deviates from 0 as the lead turns, the error becomes larger; finally, as both aircraft line up in the same direction after the turn the error becomes smaller again. The opposite behavior is true for the heading error. After the initial transient period, the heading error is relatively large; as the bearing becomes more oblique, the heading error becomes smaller; and finally, when both aircraft line up, it becomes larger.

(ii) The standard deviations for the ground speed were 13.0, 21.5, and 41.7 knots, and the average peak mean errors were 19.4, 21.3, and 30 knots for nominal, twice and four times error levels, respectively.

The standard deviations for the heading were 5.8, 10.3, and 19.9°, and the peak mean errors were 5.4, 6.3, and 8.7°.

The standard deviations of the heading error are roughly proportional to the sensor-error level. The ground speed error shows a trend, but it is not as clear-cut as for the heading errors. The peak errors are similar; for the nominal and twice nominal cases. They are 19.4 vs 21.3 knots and 5.4 vs 6.3°, but at four times the nominal, the peak mean errors increases almost 50%.

(iii) If the numerical value of the target ground speed is to be shown as a part of target-data tag, then it may be advisable to quantize the magnitude so that pilots are not annoyed by the noisy digital indication.

(iv) The third row of the summary Table 16 shows the 60 sec projected position as based on 200 knot target speed. The table shows the standard deviation and the maximum rms magnitude. Accordingly, for example, in the case of the nominal sensor error level, the tip of the target prediction vector would, on the average, move $\pm$0.4 nmi ($\pm 1\sigma$) from one sample period to another. This implies that, at a display scale of 1 nmi/inch, the vector tip hops around 0.4 inch.

This indicates that the digital display of the ground speed and the target prediction vector may be pilot selectable CDTI functions, i.e., the pilot might choose to suppress the display elements if the signal quality becomes below his acceptable level.

Table 16. Summary of Prediction Vector Error for Non-Aided x y Tracker

	Nominal	Twice Nominal	4 Times Nominal
Ground-speed Error knots	13.0(1) (19.4)(2)	21.5 (21.3)	41.7 (30.0)
Heading Error degrees	5.8	10.3	19.9
Prediction Position Error (60 sec) nmi	0.4(3) (0.49)(4)	0.70 (0.86)	1.35 (1.53)

(1) Average standard deviation.
(2) Average peak mean error.
(3) Standard deviation of 60 sec prediction position error.
(4) Maximum rms 60 sec prediction position error.

The following are tentative conclusions. It should be noted that they are based on simulation results of one particular scenario, and the TCAS/CDTI sensor is the only error source.

(a) The target ground speed and heading angle estimates may provide useful CDTI information if the sensor error magnitudes are less than $\pm$ 150 ft ($\pm 1\sigma$) and $\pm 2^\circ$ ($\pm 1\sigma$) for

the range and bearing, and if the underlying relative geometry is favorable.

(b) The target ground speed and heading-related CDTI display parameters, such as the ground-speed data tag or the target prediction vector, may (should) be made pilot selectable functions. Pilots would be able to decide at what noise level these signals cease to become useful for their tasks.

The conslusions are supported to a certain extent by a simulation study. A NASA study [24] showed that the pilot's intrail-following performance did not deteriorate much (qualitatively speaking) for up to 20 knots of target ground-speed indication.

Application of Smoothing Algorithms. In some CDTI applications, the state estimates at some past time are more important than those at the current time. This is the case with the so-called Constant Time Delay in-trail following task. Very briefly, this criterion states that the Own follows the Lead T_D seconds later. Mathematically, it can be expressed as:

$$d_0(t) = d_T(t-T_D) \tag{101}$$

where $d_0(.)$ and $d_T(.)$ are the distances traveled by Own and the Lead along a fixed air route, and T_D is a fixed delay time. The current Own position is where target was T_D sec ago. One of the major advantages of this criterion compared to others (for example, a Constant Time Predictor) is that the velocity profile must be identical except for the time delay, i.e.,:

$$v_0(t) = v_T(t-T_D) \ . \tag{102}$$

Reference [25] discusses analytical aspects of the various in-trail following criterion.

From the regulator control viewpoint, the perturbation acceleration command, δa_c, to satisfy the criterion of Eq (101) may be expressed as:

$$\delta a_c(t) = K_p[d_0(t)-d_T(t-T_D)]+K_D[v_0(t)-v_T(t-T_D)] \tag{103}$$

where K_P and K_D are the proportional and derivative regulator gains.

Equation (103) may be a basis for a flight director design via a speed-error tape indicator, or an outer-loop guidance law design; however, it is not advocated that Eq (103) is implemented exactly. Exact implementation would depend on other factors such as the inner-loop design.

It is interesting to obtain a rough idea of sensor noise effect on the acceleration command error. The following are assumed:

position error = 60 ft (1σ);
speed error = 13 knots (1σ);
proportional gain (K_p) = $(0.2)^2$ sec $^{-2}$; and
derivative gain (K_D) = 0.2 $\sec^{-1}$.

Then the acceleration command error standard deviation, σ_a, is given by:

$$\sigma_a = [Kp^2\sigma_d{}^2 + K_D{}^2\sigma_v{}^2]^{\frac{1}{2}} = 2.6 \text{ knots/sec} \tag{104}$$

Obviously, this would be excessive. Most of the above error is attributable to the velocity error.

In the majority of the CDTI symbology devised for simulation studies, the pilots extract the necessary control parameters from the history dots with one exception [26]. In the exception case, a flight director type display was also used in conjunction with the history dots. The Lead past positions are displayed as dots. The one corresponding to $T_D$sec ago is marked specially, and the target ground speed alphanumerics may be shown by the special symbol as the speed reference. The pilot tries to put Own aircraft symbol on the dot with the proper speed to satisfy the criterion.

Those display parameters (especially the ground speed) were generated based on simulated true values. However, in actuality, these must be based on the estimates. There are essentially two methods of generating the position and velocity estimates, $d_T(t-T_D)$ and $v_T(t-T_D)$. One is to use the filtering algorithm, i.e.,

$$E\{d_T(t-T_D) \mid \text{measurements up to time } t-T_D\}. \tag{105}$$

The other is to use the smoothing algorithm, i.e.,

$$E\{d_T(t-T_D) \mid \text{measurements up to time } t\} \tag{106}$$

Here, the notiation E{ } means the standard conditional expectation. A smoothing algorithm of particular interest is the so-called fixed time lag, fixed interval smoother [27]. The fixed time lag means that only the estimates at time $t-T_D$ are computed as the current time, t, advances. The fixed interval means that the data interval is fixed rather than extending all the way to the initial time. Usually the fixed interval is taken symmetric with respect to the reference time, $t-T_D$.

Appendix C derives a proposed recursive fixed time lag, fixed interval smoothing algorithm based on a linear three states Newtonian dynamic model. Appendix D derives a proposed nonlinear smoother algorithm based on a circular-arc trajectory dynamics model. It incoporates a parameter identification subalgorithm based on the bank-of-Kalman-filters idea. Readers are referred to the Appendices for detail.

During the course of this study, these algorithms were implemented to test their applicability. The following comments pertain to this effort.

(i) The recursive algorithm experienced a numerical stability problem. One cause of this problem was thought to be the fact that the closed form poles (i.e., eigenvalues of the system matrix, F, in Eq. (C.27) are all 1. However, a semirecursive least-square smoother algorithm based on Eqs (C.13), (C.23), and (C.8) did not experience numerical instability problems. In this approach, the 3x3 matrix inverse may be precomputed and stored.

(ii) Because of the modeling errors during the turn maneuver, the usable smoothing interval was 7-11 sec, (i.e., half intervals of 3-5 sec at the nominal error levels). With these short smoothing intervals, the estimates were not much more accurate than the fixed gain filtering algorithm.

(iii) Two methods to use nonlinear interpolation algorithms were attempted as described in Appendix D. One was based on Fig. D-1, and the other was a linear least squares identification algorithm based on Eq (D.11).

(iv) Because of the dynamic delay problem associated with the step change in turn-rate, ω, the smoothing interval was short compared to the linear case.

(v) Major problems associated with the bank-of-Kalman-filters decision process (Fig. D-1) were that the individual rms error distribution had more than one local minima. Furthermore, these minimum rms errors showed substantial random nature from sample to sample.

(vi) The effort to identify the unknown coefficient ($\cos \omega\Delta$) in the auto regressive equation (D.11) using a standard technique [28] was not effective. The sensor did not

provide sufficient precision to identify the value of $\cos \omega\Delta = 0.9986$ for $\omega = 3°/sec$. In order to obtain the statistical precision of this order, one would have to use thousands of measurements during which ω would have changed.

The basic conclusions from the smoother study are two: (1) smoothing algorithms are of limited value for the CDTI applications, and (2) it is not possible to estimate (filtering or smoothing) the turn rate from the TCAS measurements. Compared to the previous results on the position and velocity estimates based on the Mode S ground sensor [3], the possibility of obtaining usable turn rate estimate is very small for the TCAS sensor. The differences are 0.04 vs 1° bearing error magnitudes and 4.6 vs 1 sec sampling times for the ground-based system and the TCAS.

From the data storage point of view, to generate the past $T_D = 60$ sec history dots at 4 sec apart, the smoother algorithm required substantially more memory--at least 280 cells compared to 60 cells for the filtering algorithm. For the filtering algorithm, the target x y position and velocity estimates (with respect to the underlying map) need to be stored every 4 sec. For the smoothing algorithm (assuming a 10 dots ahead and 10 behind algorithm), the last 70 sec worth of relative range and bearing measurements as well as pseudo x y measurements need to be stored. The relative range and bearing are used to compute measurement error covariances. The computational requirement for implementing the smoother algorithm is an additional load.

Concluding Remarks

In general, the CAS logic requires a higher accuracy in state estimates as compared to the CDTI applications. The difference is functional. TCAS must generate an advisory the pilot may follow based on the available information, whereas CDTI provides an information base to the pilot so that he can make decisions, thus, the former is more tactical whereas the latter is more strategic. As such, the TCAS function can be gauged in a quantitative manner--for example, the threat detection reliability can be related to the estimation accuracy. On the other hand, the measure of merits for the CDTI functions are from the pilot utility viewpoint. Thus, for the CDTI applications, the estimation accuracy analysis is relative (in the sense

of human factors research) rather than absolute. It is the subjective opinion of this author that a reasonable and accurate traffic sensor (say, one with less than 6° bearing error) would provide an accurate enough strategic information base to the pilot.

VI

CONCLUSIONS

In this report various TCAS sensor estimation problems were examined from the viewpoint of CDTI and CAS applications. The enhanced TCAS II designed by Bendix was used as the traffic sensor basis; however, parts of the analysis are also applicable to the minimum TCAS II traffic sensor. Three problem areas were investigated - horizontal x-y, range and altitude estimation. The insight obtained from this study is summarized below.

In Chapter II, horizontal x and y estimation algorithms were developed and analyzed concerning two main factors - signal configuration and filter gain selection. By signal configuration is meant any additional information available for complementing the basic TCAS measurement of range and bearing. Own and target accelerations (or differential velocities) were selected as generic complementation signals. The target acceleration signals were assumed to be cross-linked via Mode S data link. Three filter configurations were developed based on three combinations of different signals. These were (1) non-aided; (2) Own data-aided; and (3) Own and Target data-aided.

Three gain selection methods were developed and discussed for each configuration - fixed gains, Kalman filter gains, and table-look-up gains. (The current Bendix TCAS uses the fixed gain, non-aided configuration algorithm.) Performance analysis data were obtained with respect to TCAS sensor noise level, TCAS surveillance interval, and Own and Target maneuvers using Monte Carlo simulation method. Based on the "raw" error performance statistics, the following conclusions are appropriate:

(i) Combination of the Own and Target data-aided configuration and the Kalman gain updating exhibited the best results;

(ii) Own data-aiding helped when the relative accelerations were due to Own maneuver; however, this was not always true when Target and Own maneuvered;

(iii) Non-aided configuration had good performance when the underlying kinematics were rectilinear. It developed large and sustained velocity errors, if Own or Target maneuvered; and

(iv) Monitoring the measurement residuals for sustained (bias) errors is necessary to know when not to use the estimates. Also, see Conclusion (vi) below.

In Chapter III, several range filter algorithms were analyzed. These are range and range rate estimated based on (1) non-aided horizontal filter; (2) Own and Target data-aided horizontal filter; (3) two state $\alpha\beta$ tracker; and (4) three state range square $\alpha\beta\gamma$ tracker. Performance analysis data were obtained for each algorithm with respect to TCAS sensor noise levels and encounter geometries using the Monte Carlo simulation method. Four scenarios - were selected to simulate collision encounters: tail chase, route crossing, and head-on and parallel turn-in. Based on the "raw" estimation performance data, the following conclusions are appropriate:

(v) The aided configuration showed the best performance in all cases. The estimation errors were similar in magnitude regardless of the encounter geometry. The errors were proportional to the measurement error levels:

(vi) The non-aided configurations were better than the other two range only estimators (the range $\alpha\beta$ and the range square $\alpha\beta\gamma$ trackers) except the parallel turn-in encounter. The difficulty was the large and sustained mean range rate error caused by the filter dyramic delay;

(vii) The range square filter performed credibly for rectilinear encounters. However, this may be due to the gains being too low; and

(viii) The range $\alpha\beta$ filter suffered from the nonlinear behaviour of range at or near the minimum range even for rectilinear encounter cases. It exhibited very quick recoveries due to its high gain nature; however, due to the very same nature, it passed a large portion of the high frequency noise achieving less signal smoothing than other filters.

In Chapter IV, the vertical estimation problems were addressed. The main concern was the treatment of the 100 ft quantized Mode C altitude reports. A new algorithm called the Level Switching Time (LST) filter was designed for this purpose. It was investigated extensively comparing its results with those of an $\alpha\beta$ tracker with non-quantized altitude input using Monte Carlo simulation. The $\alpha\beta$ tracker is used to obtain Own altitude and altitude rate estimates. Thus, it represents the best performance possible without complementing it with other signals such as vertical acceleration. The following conclusions apply to this chapter:

(ix) If the altitude additive noise is correlated rather than independent, the basic αβ tracker algorithm needs to be modified;

(x) The LST tracker performance is very credible considering that it must work with the 100 ft Mode C quantization. In a low to medium rate (5 ~ 20 fps) regime the steady state LST estimates are smoother. In the high speed regime (~ 60 fps), the LST estimates are somewhat inferior. The peak mean errors caused by the time delay due to the 100 ft quantization are larger;

(xi) The basic dynamic delay problem with the LST tracker remains the fundamental problem. Compared to the αβ tracker, the delay represents an extra ~ 20 sec at 5 fps and ~ 6 sec at 60 fps. However, this may be a limit which cannot be solved by algorithmic considerations alone. For example, it may require a cross-link or other signals such as the vertical acceleration.

Various estimates are used to calculate dynamic parameters for other applications. In Chapter V, the "raw" estimation performance statistics were used to infer impacts to selected CAS and CDTI applications. These applications include:

Surveillance Function

- range sampling schedule;
- altitude sampling schedule;

Collision Avoidance Logic

- range closure time;
- relative altitude projection;
- one vs two dimensional horizontal threat detection;

Cockpit Display Applications

- bearing errors;
- target prediction vector.

Both linear error analysis and Monte Carlo simulation methods are used to draw technical conclusions. Additionally, smoothing (as opposed to filtering) algorithms were investigated for an active CDTI mode application. The algorithms are developed in Appendices C and D. The following conclusions are tentative to the extent that the statistical base is limited.

(xii) The next surveillance schedule time computations based on either range rate or altitude rate (LST) estimates were affected very little. The maximum deviation was 22% for the range αβ tracker results with the four times the nominal noise level. However, the results are applicable to estimates generated with 1 sec sampling interval only;

(xiii) For the range closure time (tau) computation, all the tracker configurations performed well for rectilinear encounter kinematics. The worst was 6.5 sec (22% of 30 sec protection time) error for the αβ tracker for the head-on encounter case when the measurement noise was four times the nominal. For this case, 34% of the time, threat warning would be delayed 6.5 sec.

The closure time errors were 4.2 to 6.3 sec for the parallel turn-in case for all except the aided tracker configuration. "Half" of the errors were attributable to the dynamic delay. Therefore, this error would be sustained for the duration of the maneuver;

The closure time errors would be larger during the initial transient period. This implies that the CAS protection would not be reliable for pop-ups;

(xiv) The error in projected relative altitude 30 sec into future could be less than 200 ft for steady climb rates for Own and target and between 240 to 880 ft during the maneuver transients. Compared to the altitude separation threshold of 850 ft, the former error magnitude would be satisfactory and the latter would not;

If the combined (Own and target) low frequency error (bias, scale factor, or drift) is 200 ft or larger, then the vertical threat assessment accuracy becomes marginal;

(xv) Two dimensional (x and y) threat assessment test was found to be superior compared to the one dimensional (range only) test; for the head-on encounter case, the threat assessment using the horizontal miss-distance was 100% accurate at the range of 2.7 nmi (30 sec to CPA) for the nominal measurement noise level. The reliability dropped to 65% for the non-aided tracker if the noise level was four times the nominal;

The threat assessment became more accurate as the range became closer;

(xvi) For the CDTI applications, relative bearing should be computed within the stabilized local level reference frame. If this is not done with the TCAS sensor, this should be done using an onboard (navigation) computer. The error could be as large as 5 deg for the Own roll attitude of 20 deg;

(xvii) The non-aided configuration velocity errors in terms of ground speed and heading were quite large for the CDTI station keeping task. This is true for a maneuvering lead aircraft. The errors ranged from 23-51 kt rms for the ground speed, and 5-20 deg rms for the heading as the noise level was increased to four times the nominal;

The above numbers translated to 0.6 - 2 nmi rms excursions of the tip of the 60 sec target prediction vector;

For generating the prediction vector with non-aided filter configuration, the range and bearing errors should be better than $\pm$ 150 ft ($\pm 1\sigma$) and $\pm$ 2 deg ($\pm 1\sigma$), respectively, if the underlying kinematics are rectilinear.

Basic conclusions from the smoother algorithm study effort are two:

(xviii) Smoothing algorithms are of limited value for the CDTI applications. This is because the smoothing algorithms were limited to operate on relatively short intervals due to the dynamic (turn) consideration; and

It is not possible to estimate turn rate from the TCAS measurements. The signal-to-noise ratio is too high for the required precision.

APPENDIX A

Brief Functional Description

of

Enhanced TCAS II Traffic Sensor

NASA Langley Research Center is pursuing a research effort concerning the Cockpit Display of Traffic Information (CDTI) concept. The CDTI is a device which presents information to the pilot and crew depicting the status of surrounding traffic including position and velocity states. The traffic information is provided by a "traffic sensor." Because there seems to be no official impetus to develop a CDTI traffic sensor per se at this time, an experimental sensor must be developed based on related systems which are currently being developed. The FAA developed Traffic Alert and Collision Avoidance System (TCAS) comes closest to fulfilling various CDTI research needs.

TCAS is strictly an airborne system which provides the aircraft separation protection information independent of the ground ATC system. The FAA plans call for developing two types of TCAS -- TCAS I and TCAS II. Within each category, a certain latitude in capability is allowed to satisfy a wide spectrum of user requirements. The enhanced TCAS II which is capable of obtaining relative bearing measurements between the protected (Own) and surrounding aircraft (Target) may be able to support CDTI applications. There are two designs in this enhanced TCAS II category. One design developed by MIT/Dalmo Victor is based on the so-called active Beacon Collision Avoidance System (BCAS). The other developed by Bendix is based on the so-called full BCAS concept. Table A-1 shows the over-all performance and operational characteristics of these two systems.

The enhanced TCAS II is capable of range and bearing (in addition to the encoded altitude) measurements with a medium degree of accuracy to the extent that a more sophisticated CDTI type display or horizontal collision avoidance logic may be supported. Table A-2 shows the consensus of engineering opinion indicating the TCAS functional breakdown and bearing accuracy.

Table A-1. Operational Characteristics of Candidate CDTI Sensors

	DV TCAS II	BX TCAS II
Mode Base	• Aircraft based • Independent of ground	• Aircraft based • Independent of ground
Coordinate System	• Aircraft fixed • Aircraft Body Axes	• Aircraft fixed • NED Local Level
Coverage	• Within 10 nmi (max) of Own	• Within 25 nmi (max) of Own
Accuracy	• range = 100 ft rms • bearing = 6-8 deg rms • altitude = 100 ft resolution	• range = 100 ft rms • bearing = 2 deg rms • altitude = 100 ft resolution
Sampling Rate	• 1 sec • reliability = f(range)	• 1-8 sec (variable) • reliability high
Comments	• αβ tracker in r-β axes • LO tracker (altitude) • Estimate not stabilized with respect to Own attitude • Global protection . Medium traffic density	. αβ tracker in x-y axis . LO tracker (altitude) . Estimate stabilized with respect to Own attitude, i.e., ϕ, θ, ψ are available within the TCAS processor . Global protection . High traffic density

Table A-2 Functional Breakdown of TCAS II with respect to Bearing Accuracy

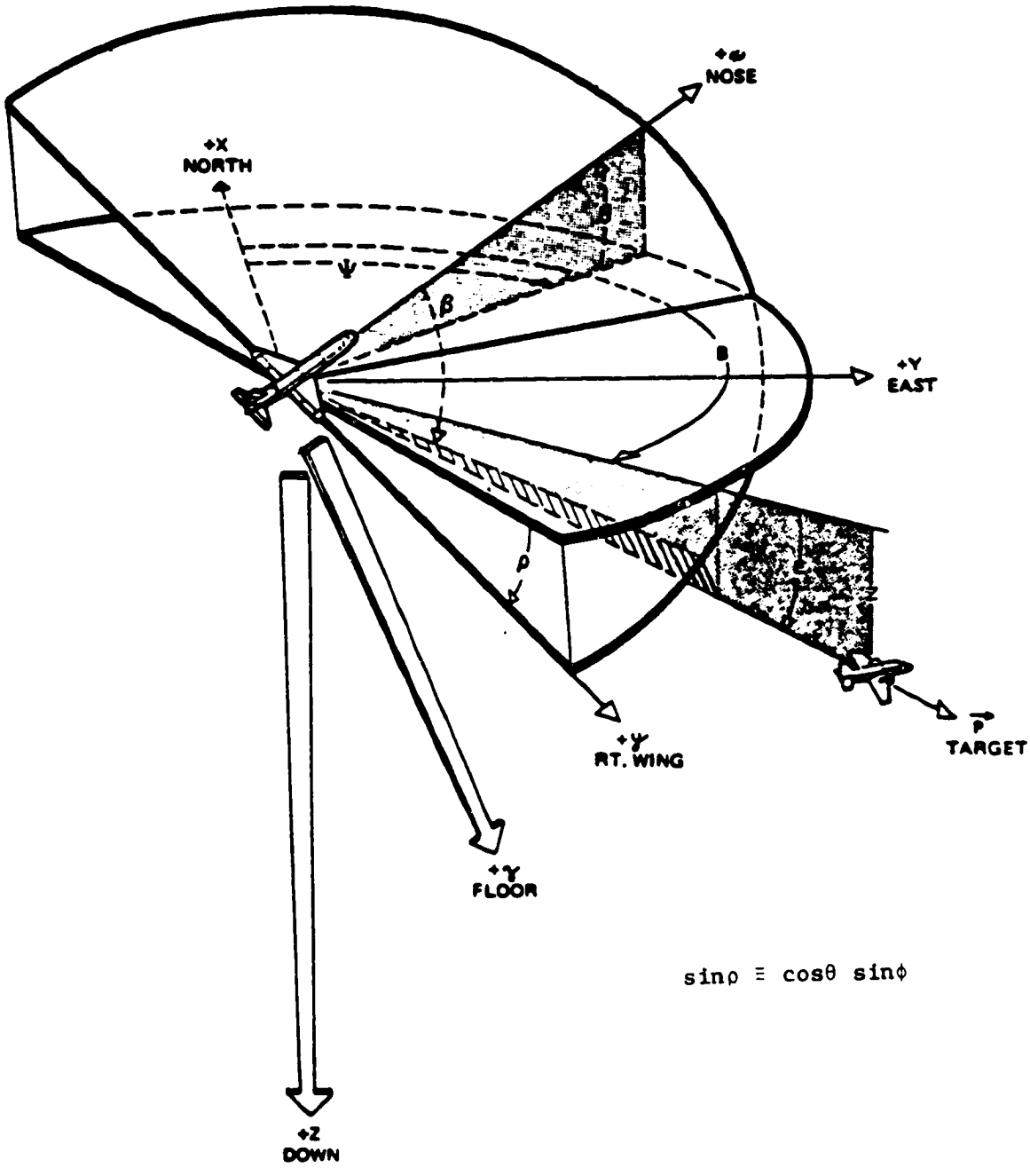
	Bearing Accuracy (deg)	Function
Enhanced TCAS II	4 - 8	o Vertical Resolution (bearing modified) o PWI or CDTI
	0.6 - 2	o Horizontal and Vertical Resolution o CDTI

In the subsequent sections of this appendix, a brief functional description of the FAA/Bendix enhanced TCAS II traffic sensor is given. This type of system seems to be more suitable for the CDTI applications in terms of coverage volume, accuracy and versatility.

Coordinate Systems Two coordinate systems are important in TCAS sensor geometry. One is a north referenced local level coordinate system attached to the Own fuselage at the antenna. The other is the orthogonal coordinate system attached to the antenna plane, i.e., the aircraft body reference system. Figure A-1 depicts the transformation geometry.

The relative bearing is measured with respect to the latter reference (the relative range is coordinate free); whereas the relative position (say, north-east-down) is measured in the local level system.

Using the conventional definition of Euler body angles ϕ, θ and ψ, the transformation from the local-level to Own body axis (T_{BL}) is given by



$\sin\rho \equiv \cos\theta \sin\phi$

Figure A-1. TCAS Geometry

$$T_{BL} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix} \qquad (A.1)$$

Using this transformation, relative north-east-down position vector to a target aircraft transforms to that of body axis as

$$\begin{bmatrix} \Delta x_B \\ \Delta y_B \\ \Delta z_B \end{bmatrix} = T_{BL} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} . \qquad (A.2)$$

And the relative bearing and elevation angles to the target are given by

$$b = \tan^{-1} (\Delta y_B / \Delta x_B) ,$$
$$e = \tan^{-1} (\Delta z_B / \Delta r) . \qquad (A.3)$$

Target Track Establishment The function of establishing the target track consists of two subfunctions: relative position measurement and the associated target correlation. The position measurement refers to the actual RF (Radio Frequency) activities between Own's transmitter/receiver and Target's transponders and the subsequent signal processing to extract the position measurement. The correlation process, also referred to as the track acquisition or establishment, establishes the correspondence between a set of measurements and a particular tracked aircraft.

The surveillance process begins by the TCAS transmitting 1030 MHz interrogation signals and by receiving 1090 MHz replies from nearby transponders (Mode A, ATCRBS or Mode S) or by listening for Mode S squitter or air-to-ground transmission signals at 1090 MHz. The positional measurements are then computed by the internal signal processor as follows:

(a) range--by the time duration between the interrogation and the corresponding reply reception, accounting for the transponder delay;

(b) bearing--by computing the angle-of-arrival from the phase distribution among several antennas; and

(c) altitude--by decoding the Mode C altitude code contained in the reply. (For a Mode A only target, this will be non-existent.)

The surveillance characteristics of the BX TCAS are somewhat similar to that of the ground based Mode S beacon sensor. Because a large number of transmitters in a small locale will cause interference resulting in synchronous garble, fruit or false squitter detection, there are three techniques (in addition to the mono-pulse technique) to overcome the high density problem. One is the interrogation antenna directivity; the second is the so-called "whisper/shout" signal power level sequencing; and the other is the interrogation rescheduling if a reply is missed or garbled. The antenna beam width is 22½ deg; however, by repeating the transmission four times and each time sliding the beam center by 5.625 deg, the effective beam width becomes 5.625 deg. The beam pointing and rescheduling as well as several levels of whisper/shout power sequencing are controlled by internal digital processors based on the internal track file, Own aircraft orientation, and ATCRBS/Mode S transponder mix. The task is facilitated by the fact that the beam is "stabilized" with respect to roll, pitch and yaw attitude angles.

It is simple to track Mode S equipped aircraft because of the uniquely assigned discrete address in the reply format (which is also stored in the TCAS unit). The task of correlating between the measurements and the tracked aircraft is not as simple if the target is Mode C or Mode A equipped. Also, for the narrow beam system (BX TCAS), the correlation process is simpler than for the omni-directional system, because the number of replies corresponding to an interrogation is generally much smaller. However, even the narrow beam width and the rescheduling capability present problems if two or more aircraft are clustered in close proximity.

A "gating" technique is used for the purpose of separating targets. If the current measurement falls within certain threshold values (which define the gate) of an aircraft in the track file, then the measurement is assigned to that aircraft, and the corresponding track file is updated. If the measurement does not correspond persistently (5-10 sec) within a gate to any existing aircraft in the track file, then a new track file is started for that aircraft. Conversely, if none of the measurements consistantly correspond to an aircraft in the file, then that aircraft is judged outside the beam reach, and hence, it is deleted from the track file.

Coverage Volume and Interrogation Scheduling Logic The Own protected airspace provided by the system is physically limited because of the device's power output limitation. Also, the beam pattern due to the antenna configuration comes into play, especially for the vertical coverage. The maximum beam reach is estimated to be 35 nmi; this is at the highest sensitivity level. Within this distance, the 1030 MHz transmission signals can be distinguished from the ambient radio frequency (RF) noise with a certain reliability. On the other hand, the vertical limitation is due to the elevation beam shape. The mounted antenna assembly is designed to provide coverage of approximately five (5) deg below and 23 deg above the antenna plane. The system may or may not include a similar antenna assembly located at the bottom of the fuselage.

To limit unnecessary RF activity, especially in a high traffic density area, the Bendix system relies on an "artificial" boundary generated by the beam control microprocessor. The volume is dynamically computed and is defined by the relative range and Own altitude. In the Case of Mode A only transponder, the range defines the volume. Furthermore, the volume is subdivided into two regions - "acquisition" and "track".

The acquisition region is provided mathematically by

$$\Delta r \leq \Delta r_{acq} , \qquad \text{and} \qquad |\Delta h| \leq \Delta h_{acq} .$$

Here Δr_{acq} is the (acquisition) range threshold (nominally 25 nmi), and Δh_{acq} is the altitude threshold (nominally 6000 ft).

The track region is provided mathematically by

$$\Delta r \leq \Delta r_{trk} , \qquad \text{and} \qquad |\Delta h| \leq \Delta h_{trk} .$$

Here Δr_{trk} is the (track) range threshold which is computed dynamically, and Δh_{trk} is the altitude threshold given by

$$\Delta h_{trk} = 3750 + |\dot{\hat{z}}| \cdot 45 \quad \text{(ft)} . \tag{A.4}$$

The quantity Δr_{trk} is determined based on the relative bearing as well as Own ground speed and altitude. The equations for this term are given by

$$\Delta r_{trk} = \begin{cases} \text{Max}\{5 \text{ nmi}, (\hat{V}_0 \cos b + 250)T + \Delta r_s\} & \text{for } \hat{z}_0 \leq 3000 \text{ ft}, \\ \text{Max}\{5 \text{ nmi}, (\hat{V}_0 \cos b + \hat{z}_0/20 + 100)T + \Delta r_s\} & 3000 < \hat{z}_o \leq 10000 \text{ ft}, \\ \text{Max}\{10 \text{ nmi}, (\hat{V}_0 \cos b + 600)T + \Delta r_s\} & \text{for } \hat{z}_0 \leq 10000 \text{ ft} \end{cases} \tag{A.5}$$

Here,

$\hat{z}_0$ = Own altitude, in ft,
$\hat{V}_0$ = Own ground speed, in kt,
T = "closure" time constant = 1/80 hr = 45 sec, and
b = relative bearing with respect to Own's body axes
= $\tan^{-1}(\Delta y_B / \Delta x_B)$
Δr_s = 1.05 nmi

Figures A-2 show the track regions corresponding to three Own ground speeds at three Own altitude levels.

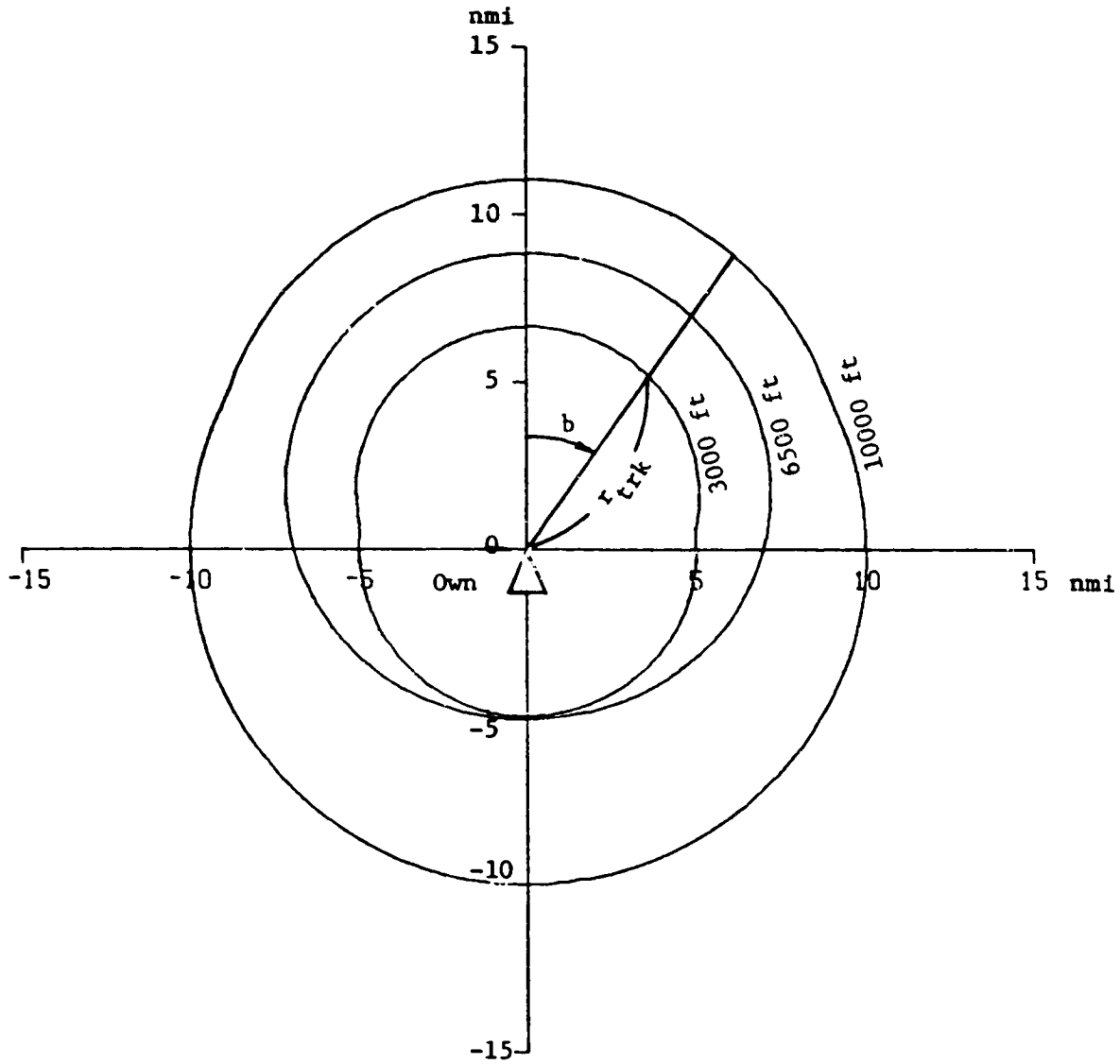


Figure A-2a. Range Track Region for V_0 = 150 kt at Various Own Altitude.

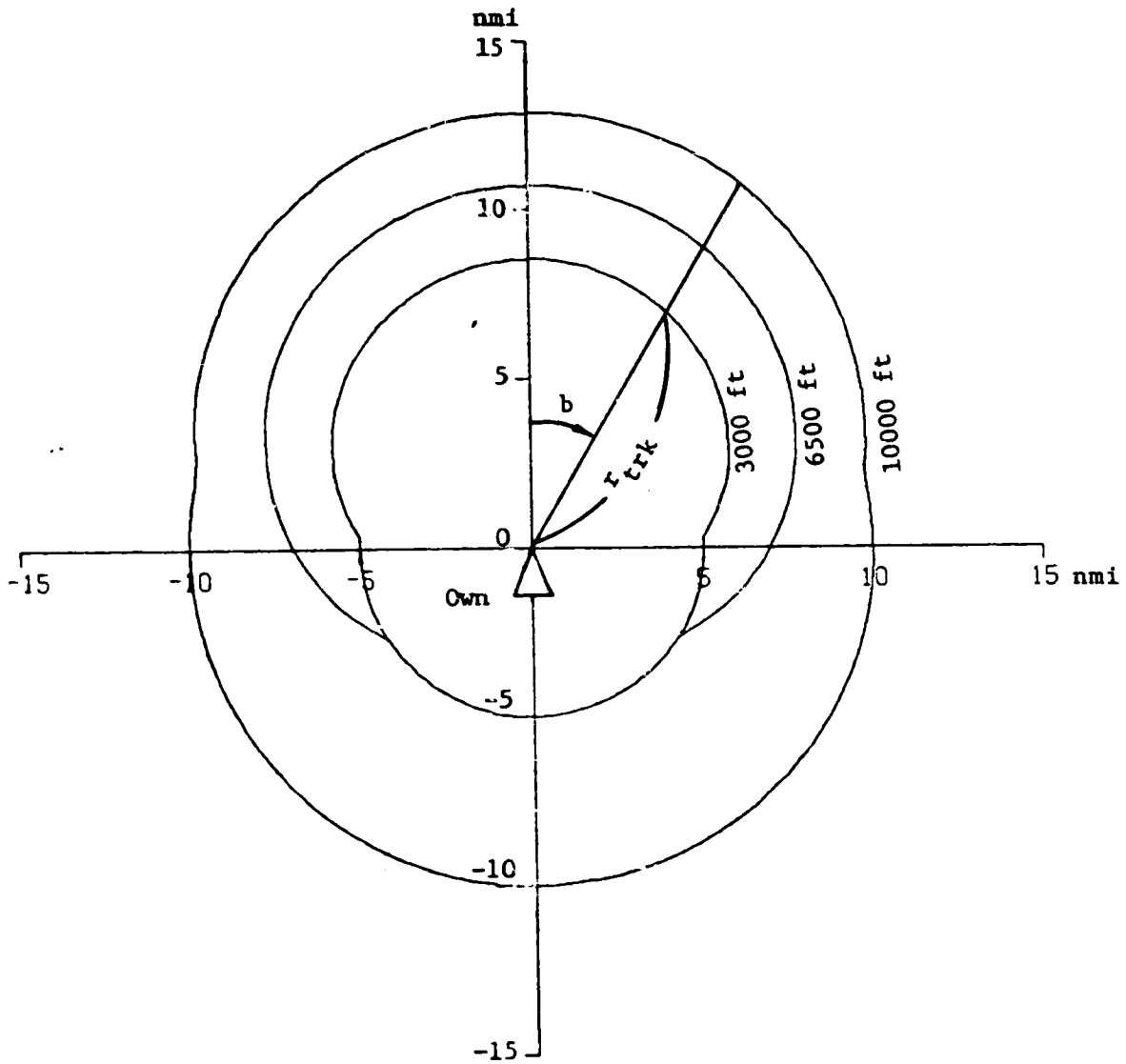


Figure A-2b. Range Track Regions for V_0 = 300 kt at Various Own Altitude.

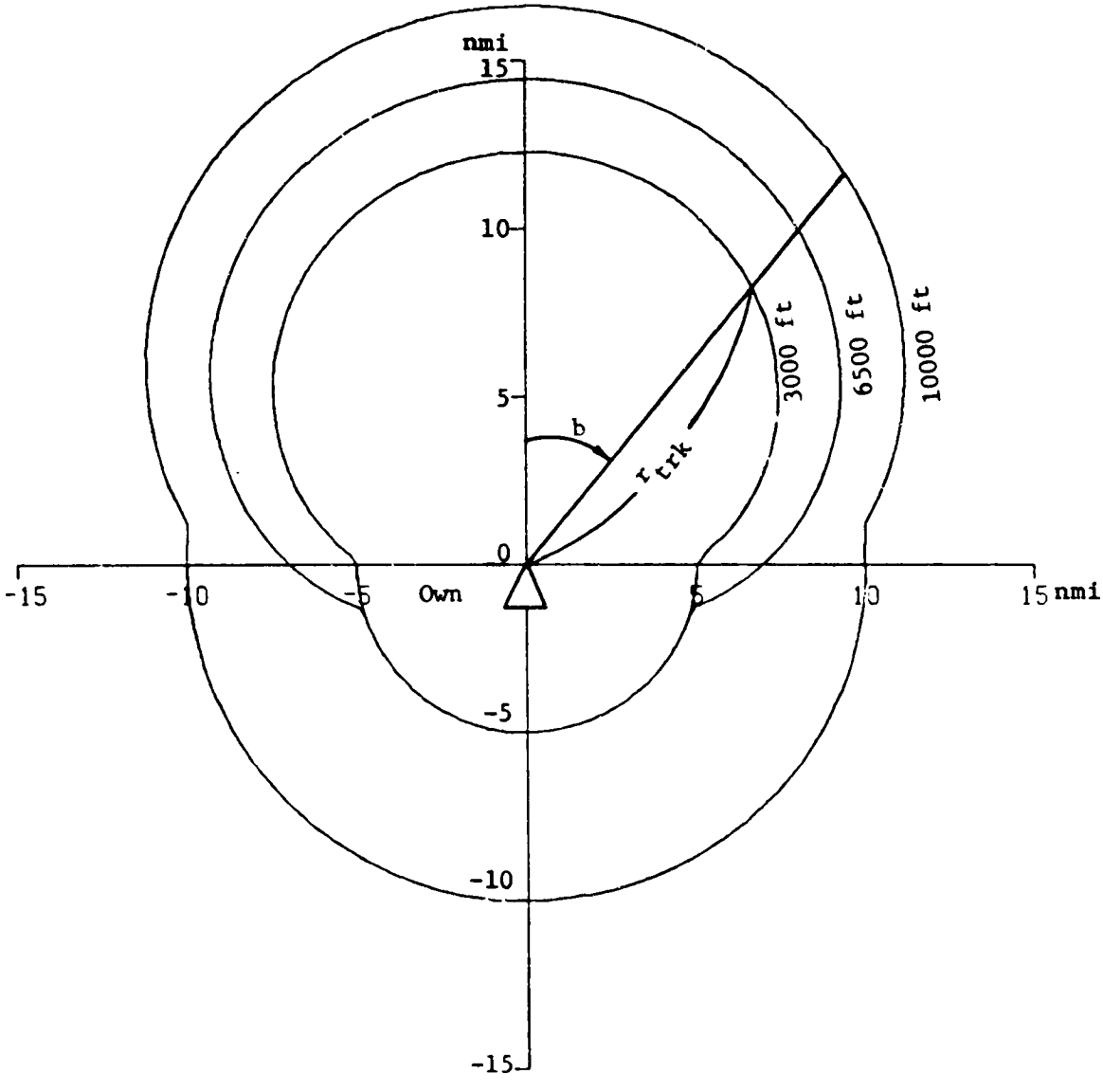


Figure A-2c. Range Track Region for V_0 = 600 kt at Various Own Altitude.

The antenna pointing controller module schedules the interrogation and reception timing (i.e., surveillance scheduling.) The surveillance operation depends on two factors. One is the transponder type - Mode C or Mode S. The other is the operational mode - search/acquisition or track. The antenna dwell at a given azimuth angle is divided into one passive and three active processes. The active ones include: (a) ATCRBS transmissions to search for new targets (targets which are not in the internal track file); (b) ATCRBS transmissions for tracking existing targets (existing in the internal track file); and (c) Mode S transmissions for tracking Mode S equipped targets. The passive process consists of possibly listening for Mode S squitters or Mode S replies to ground interrogations.

The time interval between the ATCRBS search interrogations is computed according to the formula:

$$\Delta t_s = \min \left\{ 16 \text{ sec}, \quad \frac{3600 \, \Delta r_s}{V_{Max} + \hat{V}_0 \cos b} \right\} , \tag{A.6}$$

where

V_{Max} = maximum allowed target speed,

$$= \begin{cases} 250 \text{ kt}, & \hat{z}_0 < 3000 \text{ ft}, \\ \frac{\hat{z}_0 - 3000}{20} + 250 \text{ kt}, & 3000 \leq \hat{z}_0 < 10{,}000 \text{ ft}, \\ 600 \text{ kt}, & \hat{z}_0 \geq 10{,}000 \text{ ft}, \end{cases} \tag{A.7}$$

and the other variables have been previously defined.

The ATCRBS track interrogations are made for those targets lying inside the track volume, i.e.,

$$r \leq \Delta r_{trk} \text{ .AND. } |\Delta h| \leq \Delta h_{trk} .$$

The ATCRbS track interrogation time interval, Δt_T, is computed based on the predicted relative motion of the target. It is given by the formula

$$\Delta t_T = \text{Max } \{1 \text{ sec, min } \{t_1, t_2, t_3, 8 \text{ sec}\}\} . \qquad \text{(A.8)}$$

Here

t_1 = the number of seconds it will take the target to move 3 deg in bearing,

t_2 = the number of seconds it will take the target to move 1000 ft in range,

t_3 = the number of seconds it will take the target to move 250 ft in altitude.

When a new Mode S target is detected by squitter listening, it is interrogated. If it is within 25 nmi, a track is initiated. Mode S equipped targets inside the track volume are interrogated at the same rate as if they were ATCRBS targets. Those targets which are outside of the volume but within 25 nmi of Own are tracked at a regular interval of 8 sec.

If a target (either ATCRBS or Mode S equipped) is closer than 6000 ft or if it has been declared a preliminary threat by the threat detection logic, the target track update rate is 1 sec.

If replies are missing under repeated interrogation of a tracked target, the track is dropped. In addition, ATCRBS tracks will be dropped when their coasted position (position extrapolated by dead reckoning) lies outside of the track volume. A Mode S track will be dropped when the expected range is greater than 25 nmi.

Measurement Accuracy The error characteristics for the BX TCAS in an actual operational environment are virtually unknown. The following characteristics are inferred and represent a consensus of the immediate engineering community. (A proto-type model has been in flight tests since January 1984.)

Because the interrogation/reply process of this unit is similar to the Mode S ground sensor, it is reasonable to assume that the range error could be as accurate at $\pm$ 50 ft ($\pm 1\sigma$). A standard deviation of $\pm$ 75 ft ($\pm 1\sigma$) is assumed for the simulation.

The bearing error depends on the sharpness of the directional beam and the internal clocking device. It also depends on the reflection (multi-path) characteristics from various parts of the target aircraft fuselage. The consensus value for this error is between $\pm$ 0.6 and $\pm$ 2 deg ($\pm 1\sigma$) A standard deviation of $\pm$ 1 deg ($\pm 1\sigma$) is assumed.

The 100 ft quantization due to the encoding process dominates the altitude error. Twenty-five (25) feet seems to be a reasonable standard deviation number for the high frequency error; however, low frequency drift bias or scale factor errors could be substantial with up to a $\pm$ 4% scale factor error not being uncommon.

Estimation Algorithms The basic measurements obtained by the TCAS surveillance function are relative range, relative bearing and pressure altitude above MSL. The relative bearing is referenced with respect to the antenna plane which is attached to the Own fuselage. The pressure altitude is obtained from the encoding altitude report. To perform its primary function of monitoring the threat situation and of avoiding collision, the estimation algorithms are used to derive position and velocity estimates. Therefore, the CAS application dictates the estimation algorithm requirements.

In order to support the horizontal as well as vertical resolution capabilities, the BX TCAS requires the position and velocity estimates in three dimensions. To achieve the estimation accuracy, it operates with a north referenced local level coordinate system attached to Own's

fuselage. The body referenced range and bearing measurements are transformed to north-east x and y components using the direction cosine matrix computed from the roll, pitch and yaw attitude angles provided by an on-board inertial navigation system. The resulting "raw" Δx and Δy positions are used to derive the position and velocity estimates using a simple $\alpha\beta$ tracker algorithm.

Equations for the standard $\alpha\beta$ tracker algorithm are given below for the x axis. Equations for the y-axis are identical.

$$
\begin{aligned}
x_{p,n} &= \hat{x}_n + \Delta\hat{x}_n , && \text{(prediction)} \\
\tilde{x}_{n+1} &= x_{m,n+1} - x_{p,n} , && \text{(innovation)} \\
\hat{x}_{n+1} &= \hat{x}_{p,n} + \alpha\tilde{x}_{n+1} , && \text{(position update)} \\
\hat{\dot{x}}_{n+1} &= \hat{\dot{x}}_n + (\beta/\Delta)\,\tilde{x}_{n+1} , && \text{(velocity update)}
\end{aligned}
\qquad \text{(A.9)}
$$

where Δ is the time elapsed since the last valid measurement; and α and β are tracker gains. The values of the α and β gains are tuned to compensate for the variable sampling periods. The following table lists the values.

Table A-3. α and β Gain Values

Δt sec	1	2	3	4	5	6	7	8
α	0.25	0.37	0.465	0.53	0.58	0.62	0.645	0.665
β	.066	.175	.3	.431	.565	.685	.886	.91

For the purpose of the surveillance function, a very low gain vertical tracker is used. The outputs are used essentially for the target correlation process. Within the collision avoidance logic, the vertical estimates are obtained by means of a non-linear filter based on the MIT designed Level Occupancy Time tracker algorithm.

Figure A-3 is a block diagram showing the inputs to the filters and their output to the CDTI processor.

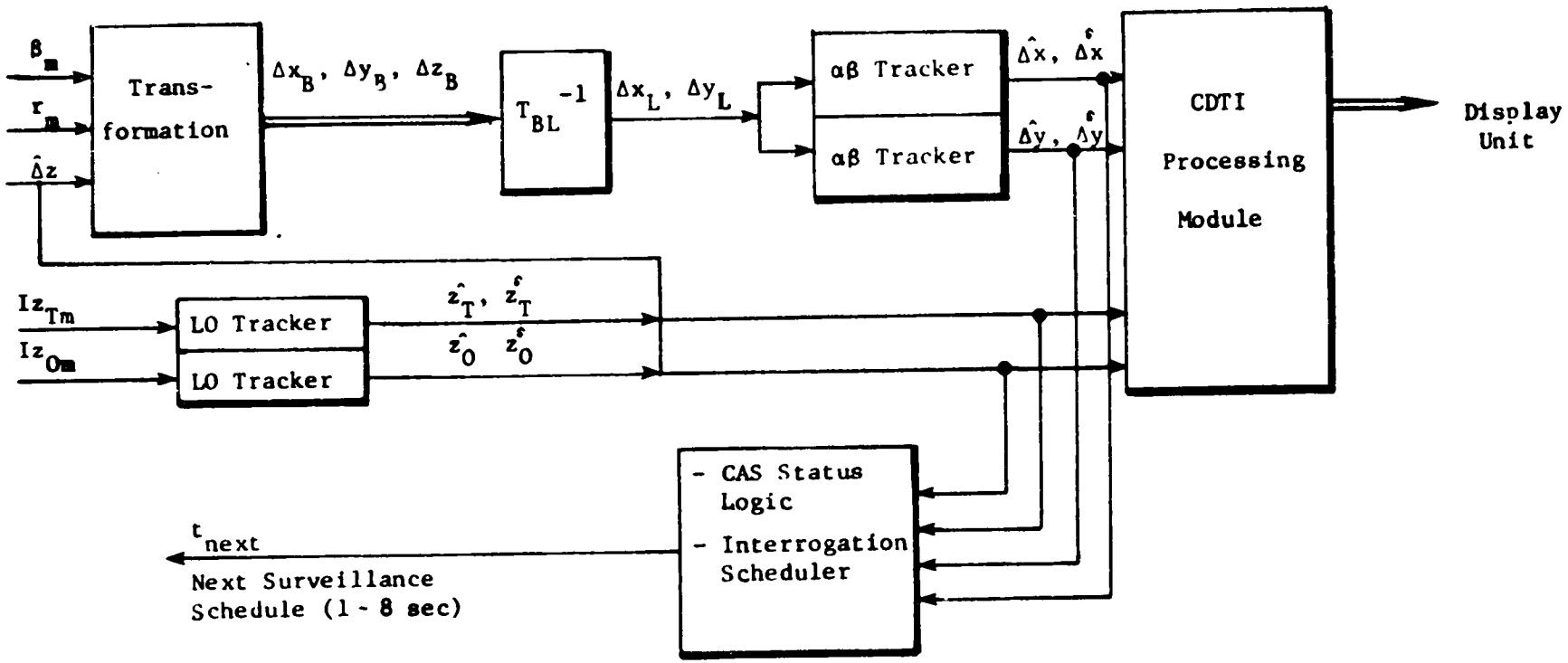


Figure A-3. Estimation, Sampler Logic and CDTI Processor Block Diagram.

APPENDIX B

Aircraft Dynamic Model

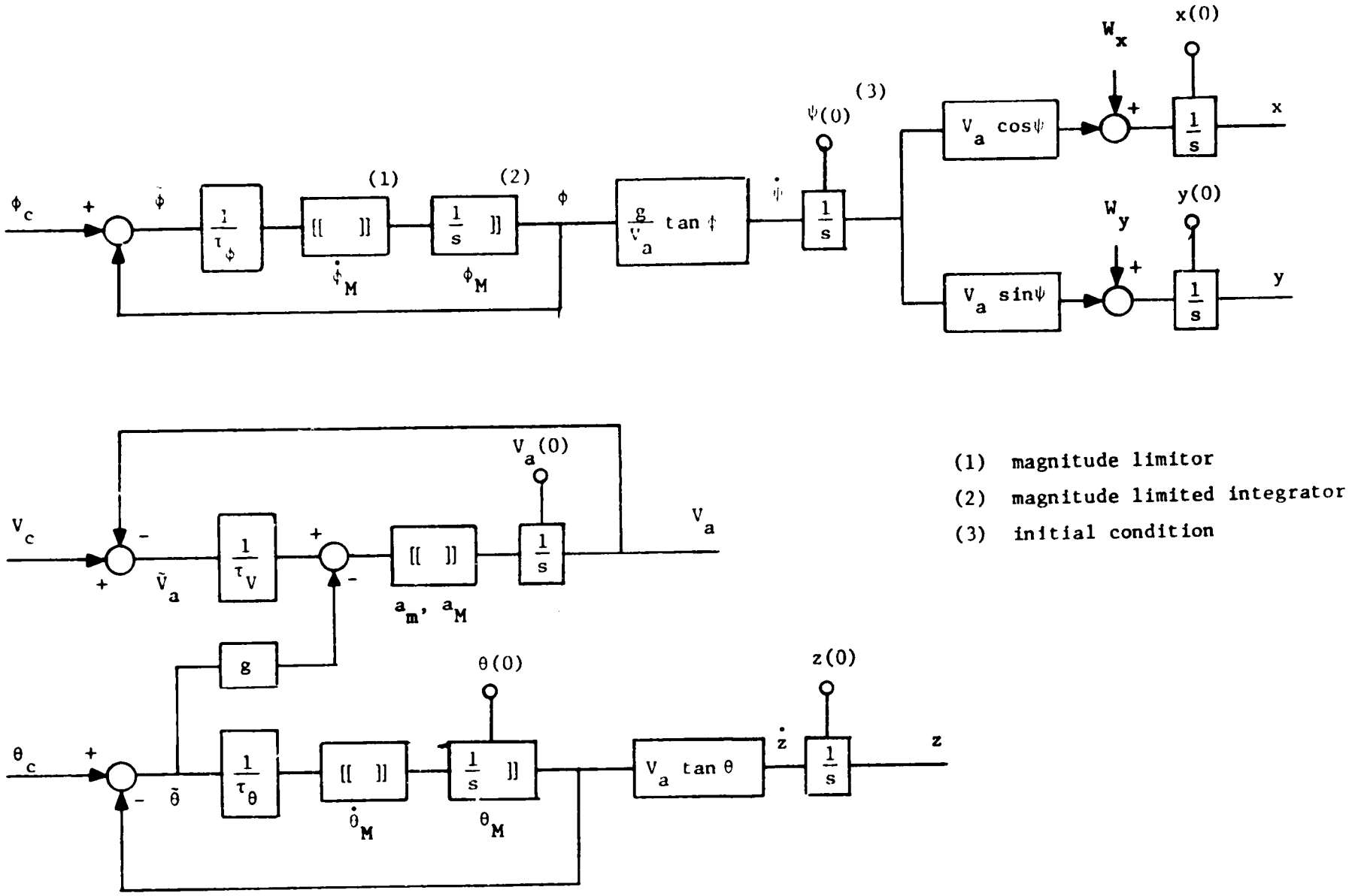
For the purpose of simulation study, the aircraft dynamic models for both Own and other traffic need to be chosen carefully. Our basic model requirements are as follows:

(a) it is simple enough for efficient computation;
(b) it includes the attitude orientation effect;
(c) it preserves the kinematics; and
(d) it represents low frequency dynamics.

Point (a) needs no explanation. Point (b) is due to the fact that TCAS measurements (range and bearing relative to Own) are with respect to the antenna plane fixed to Own fuselage. (Altitude measurement is with respect to the mean sea level via pressure altimeter.) For example, relative bearing depends not only on Own yaw angle but also on the roll and pitch angles. Furthermore, the orientation effect must preserve the kinematic relationship (point (c)), e.g., a roll angle of 15 deg at 200 knot should result in a circular arc trajectory of radius 2.2 nmi at a 1.5 deg/sec turn rate.

Because the basic sampling rate is no faster than one second, higher frequency dynamic modes are washed out by the sampling effect. Therefore, only low frequency dynamics of less than 1 sec^{-1} need to be included. These modes include the aircraft pitch and roll inner-loop closure and the throttle actuator dynamics.

A point mass, seven state, three axes model was chosen which has been successfully applied in similar studies in the past. The model is based on the inner-loop closure for pitch and roll axes and a simple airspeed select/hold law. Representative first order system dynamics are assumed between the commands (pitch, roll or airspeed commands) and the response (pitch, roll or airspeed). Figure B-1 shows the model block diagram with various authority and rate limits inserted at appropriate junctures.



(1) magnitude limitor

(2) magnitude limited integrator

(3) initial condition

Figure B-1. Aircraft Dynamic Model

The model is almost decoupled except that the pitch error ($\theta - \theta_c$) is cross-fed into the airspeed axis. This takes into account the difference between the relatively short pitch response time and the relatively long airspeed response time. The corresponding dynamic equations are given in Table B-1, and typical model parameter values are listed in Table B-2. The values are thought to be representative of civil aircraft in normal operation.

Some of the advantages of the above model over the classical model based on linearized aerodynamic forces and moments are:

1) number of state variables is smaller;
2) nominal conditions for linearization need not be considered; and
3) the model assumes a stable inner-loop, i.e., there is no need to design a control system.

The current simulation program has a provision of generating up to forty (40) aircraft including Own. The "guidance" commands ϕ_c, θ_c and V_c are generated in a deterministic way via a table look-up procedure. For example, the jth aircraft roll command, $\phi_{c,j}$, at time t is determined by the following logic.

$$\text{If } (t_{R,j} \leq t \leq t_{R,j+1}) \text{ then } \phi_{des,j} \rightarrow \phi_{c,j}$$

where the switching times, $\{t_{R,J}\}$, and the desired roll attitudes, $\{\phi_{des,j}\}$ are predetermined and stored in arrays. The other two axes are treated similarly. In this way, the model aircraft can simulate traffic with realistic dynamics. One disadvantage of this approach, however, is that the aircraft are not controlled in the 3-D or 4-D guidance sense. That is, the aircraft will not follow a predetermined track with respect to a fixed coordinate system. If such a capability is desired, the guidance commands must be generated according to the position error from a 3-D (or 4-D) path.

The dynamic equations are integrated every 0.5 sec using the trapezoidal rule.

Table B-1. Aircraft Model Dynamic Equations

roll eqn:	$\frac{d}{dt}\,\phi\,]]^{\phi_M} = [[\frac{1}{\tau_\phi}(\phi_c - \phi)]]^{\dot{\phi}_M}$	
yaw eqn:	$\frac{d}{dt}\,\psi = \frac{g}{V_a}\tan\phi$	(g = gravity constant = 32.17 ft/sec^2)
pitch eqn:	$\frac{d}{dt}\,\theta\,]]^{\theta_M} = [[\frac{1}{\tau_\theta}(\theta_c - \theta)]]^{\dot{\theta}_M}$	
airspeed eqn:	$\frac{d}{dt}\,V_a = [[\frac{1}{\tau_V}(V_c - V_a) + g\,(\theta - \theta_c)]]^{a_M}_{a_m}$	
x eqn:	$\frac{d}{dt}\,x = V_a\cos\psi + W_x$	
y eqn:	$\frac{d}{dt}\,y = V_a\sin\psi + W_y$	
z eqn:	$\frac{d}{dt}\,z = V_a\tan\theta$	
wind eqn:	$W_x = W\cos\sigma$	
	$W_y = W\sin\sigma$	

Note 1: $\frac{d}{dt}\,(\cdot)]]^{a_L}$ means that the integral $(\cdot)$ is limited in magnitude to a_L.

Note 2: $[[\,(\cdot)]]^{a_L}$ means that $(\cdot)$ is limited in magnitude to a_L.

Note 3: $[[\,(\cdot)]]^{a_L}_{a_\ell}$ means that $(\cdot)$ is limited to a_L maximum and a_ℓ minimum.

Table B-2. Typical Model Parameters

Axis	Time Constant (sec)	Authority Limit	Rate Limit
Roll	3	± 30 deg	± 10 deg/sec
Pitch	1.5	± 15 deg	± 10 deg/sec
Airspeed	6	-	1.5 knot/sec -1.0 knot/sec

APPENDIX C

Recursive Algorithm for Fixed-Interval Smoothing

Smoothing Matrix Definitions

Given the measurement data

$$\{y(n\Delta) = y_n : \quad n = -N, -N+1, .., N\},$$

the problem is to find the function value of x_n and its first and second derivatives at t=0.

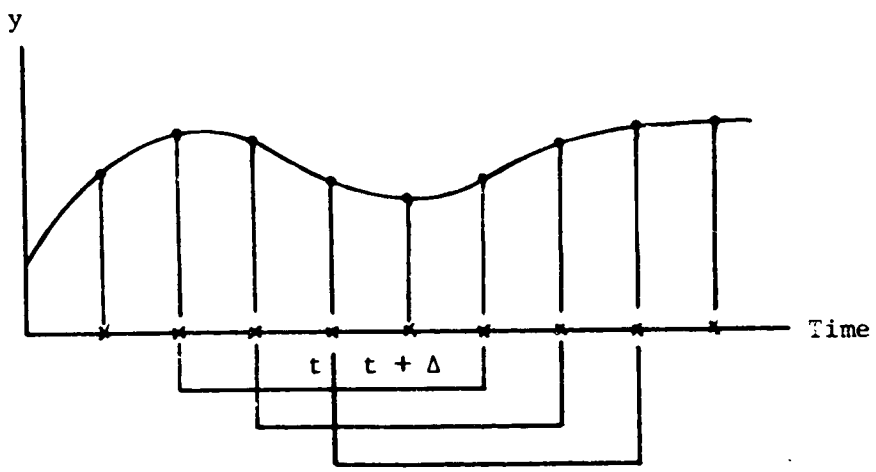


Figure C.1. Sketch of Smoothing Interval.

Under the assumptions stated below, a semi-recursive algorithm can be found for the solution. It is given by the form

$$\underline{\hat{x}}_{n+1} = \underline{\Phi}\,\underline{\hat{x}}_n + \underline{\Gamma}\,\underline{u}_{n+1} \; ; \quad \underline{u}_{n+1} = \begin{bmatrix} y_{n+N+1} \\ y_{n-N} \end{bmatrix} \tag{C.1}$$

where $\underline{\Phi}$ and $\underline{\Gamma}$ are precomputable matrices. This is based on the following assumptions

1. N is fixed, therefore, the number of data point is 2N+1;
2. The time interval between measurement points is a fixed constant; and
3. The underlying "state" equation is given by

$$\begin{aligned} \underline{x}_{n+1} &= \phi\,\underline{x}_n \\ y_{n+1} &= \eta^T \underline{x}_{n+1} + \xi_{n+1} \,. \end{aligned} \tag{C.2}$$

Let the measurement y_n be

$$y_n \triangleq y(n\Delta) = x(n\Delta) + \xi_n ,$$
$$\triangleq x(0) + n\Delta\,\dot{x}(0) + \frac{n^2\Delta^2}{2}\ddot{x}(0) + \xi_n \qquad (C.3)$$

Then we can write

$$\begin{bmatrix} y_N \\ y_{N-1} \\ \vdots \\ y_1 \\ y_0 \\ y_{-1} \\ \vdots \\ y_{-N} \end{bmatrix} = \begin{bmatrix} 1 & N\Delta & N^2\Delta^2/2 \\ 1 & (N-1)\Delta & (N-1)^2\Delta^2/2 \\ \vdots & \vdots & \vdots \\ 1 & \Delta & \Delta^2/2 \\ 1 & 0 & 0 \\ 1 & -\Delta & \Delta^2/2 \\ \vdots & \vdots & \vdots \\ 1 & -N\Delta & N^2\Delta^2/2 \end{bmatrix} \begin{bmatrix} x(0) \\ \dot{x}(0) \\ \ddot{x}(0) \end{bmatrix} + \begin{bmatrix} \xi_N \\ \xi_{N-1} \\ \\ \xi_0 \\ \\ \\ \\ \xi_{-N} \end{bmatrix} . \qquad (C.4)$$

Let

$$\underline{b} \triangleq \begin{bmatrix} 1 & & \\ & \Delta & \\ & & \Delta^2 \end{bmatrix} \begin{bmatrix} x(0) \\ \dot{x}(0) \\ \ddot{x}(0) \end{bmatrix} . \qquad (C.5)$$

Then

$$\begin{bmatrix} y_N \\ y_{N-1} \\ \cdot \\ \cdot \\ \cdot \\ y_{-N} \end{bmatrix} = \begin{bmatrix} 1 & N & N^2/2 \\ 1 & N-1 & (N-1)^2/2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & -N & N^2/2 \end{bmatrix} \underline{b} + \begin{bmatrix} \xi_N \\ \\ \cdot \\ \cdot \\ \cdot \\ \xi_{-N} \end{bmatrix} , \qquad (C.6)$$

or more concisely,

$$Y = A\,\underline{b} + \underline{\xi} \qquad (C.7)$$

The least squares solution is given by the usual

$$\hat{\underline{b}} = \left[A^T \underline{\Sigma}^{-1} A\right]^{-1} A^T \underline{\Sigma}^{-1} Y \quad , \tag{C.8}$$

where $\underline{\Sigma}$ is the measurement error covariance matrix, i.e.,

$$\underline{\Sigma} = E\left(\underline{\xi}\ \underline{\xi}^T\right) . \tag{C.9}$$

Also, it is well known that the matrix

$$\underline{P}' = \left[A^T \Sigma^{-1} A\right]^{-1} , \tag{C.10}$$

represents the covariance of estimation error, i.e.,

$$\underline{P}' = E\left(\tilde{\underline{b}}\ \tilde{\underline{b}}^T\right) \quad , \tag{C.11}$$

where

$$\tilde{\underline{b}} \triangleq \underline{b} - \hat{\underline{b}} \quad .$$

By assuming an independent and stationary nature of the error characteristics, we have

$$\underline{\Sigma} = \sigma^2 \underline{I} \quad , \tag{C.12}$$

$$\underline{P}' = \left[A^T A\right]^{-1} \sigma^2 , \tag{C.13}$$

and

$$\hat{\underline{b}} = \left[A^T A\right]^{-1} A^T Y \quad . \tag{C.14}$$

Note that $A\ A^T$ is a constant symetric matrix for fixed N.

Computation of P Matrix

By writing out the matrix multiplication from Eq. (C.6), we have

$$\underline{P}'^{-1} = A^T A = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ N & N-1 & \cdots & 1 & 0 & -1 & \cdots & -N \\ \frac{N^2}{2} & \frac{(N-1)^2}{2} & \cdots & \frac{1}{2} & 0 & \frac{1}{2} & \cdots & \frac{N^2}{2} \end{bmatrix} \begin{bmatrix} 1 & N & N^2/2 \\ 1 & N-1 & (N-1)^2/2 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & -N & N^2/2 \end{bmatrix}$$

$$= \begin{bmatrix} 2N+1 & 0 & \sum_{1}^{N} n^2 \\ 0 & 2\sum_{1}^{N} n^2 & 0 \\ \sum_{1}^{N} n^2 & 0 & \frac{1}{2}\sum_{1}^{N} n^4 \end{bmatrix} . \qquad (C.15)$$

By using the formula,

$$\sum_{n=1}^{N} n^2 = \frac{N(N+1)(2N+1)}{6} , \qquad (C.16)$$

and

$$\sum_{n=1}^{N} n^4 = \frac{N(N+1)(2N+1)(3N^2+3N-1)}{30} , \qquad (C.17)$$

Equation (C.15) becomes

$$\underline{P}'^{-1} = \begin{bmatrix} 2N+1 & 0 & \frac{N(N+1)(2N+1)}{6} \\ 0 & \frac{N(N+1)(2N+1)}{3} & 0 \\ \frac{N(N+1)(2N+1)}{6} & 0 & \frac{N(N+1)(2N+1)(2N^2+3N-1)}{60} \end{bmatrix} \qquad (C.18)$$

Taking the inverse of Eq. (C.18) gives the covariance

$$P' = \begin{bmatrix} \frac{3(3N^2+3N-1)}{(2N-1)(2N+1)(2N+3)} & 0 & -\frac{30}{(2N-1)(2N+1)(2N+3)} \\ 0 & \frac{3}{N(N+1)(2N+1)} & 0 \\ -\frac{30}{(2N-1)(2N+1)(2N+3)} & 0 & \frac{180}{N(N+1)(2N-1)(2N+1)(2N+3)} \end{bmatrix} \qquad (C.19)$$

Recursive Relationships

Define the relationship

$$q = A^T Y \quad . \tag{C.20}$$

Then, by writing out the vector equation, we get

$$q_o^1 = y_N + y_{N-1} + \ldots + y_1 + y_o + y_{-1} + \ldots + y_{-N} \quad , \tag{C.21}$$

$$q_o^2 = Ny_N + (N-1)y_{N-1} + \ldots + y_1 \quad -y_{-1} - \ldots -Ny_{-N} \quad ,$$

$$q_o^3 = \frac{N^2}{2} y_N + \frac{(N-1)^2}{2} y_{N-1} \ldots + y_1 \quad + y_{-1} + \ldots + \frac{N^2}{2} y_{+N} \quad .$$

The same equations referenced at one sampling interval later are given by

$$q_1^1 = y_{N+1} + y_N + \ldots \quad + y_{-N+1}$$

$$q_1^2 = Ny_{N+1} + (N-1)y_N + \ldots + y_2 - y_o - 2y_1 \ldots \quad - Ny_{-N+1}$$

$$q_1^3 = \frac{N^2}{2} y_{N+1} + \frac{(N-1)^2}{2} y_N + \ldots + y_2 + y_c + \frac{2^2}{2} y_{-1} \ldots + \frac{N^2}{2} y_{-N+1} \tag{C.22}$$

After algebraic manipulation, it is seen that the q_1 and q_o terms are related by

$$q_1^1 = q_o^1 + y_{N+1} - y_{-N} \quad , \tag{C.23}$$

$$q_1^2 = -q_o^1 + q_o^2 + N\, y_{N+1} + (N+1)y_{-N} \quad ,$$

$$q_1^3 = \frac{1}{2} q_o^1 - q_o^2 + q_o^3 + \frac{N^2}{2} y_{N+1} - \frac{(N+1)^2}{2} y_{-N} \quad .$$

More concisely,

$$\underline{q}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ \frac{1}{2} & -1 & 1 \end{bmatrix} \underline{q}_o + \begin{bmatrix} 1 & -1 \\ N & N+1 \\ \frac{N^2}{2} & \frac{(N+1)^2}{2} \end{bmatrix} \begin{bmatrix} y_{N+1} \\ y_{-N} \end{bmatrix} \quad , \tag{C.24}$$

$$\triangleq \quad ; \; \underline{q}_o + ! \; \underline{u}_1 \quad .$$

From Eqs. (C.13), (C.14), and (C.20), the relationship between $\underline{\hat{b}}$, $\underline{P}'$ and $\underline{q}$ is given by

$$\underline{\hat{b}} = \underline{P}'\underline{q} \quad . \tag{C.25}$$

Therefore, the recursive equation for $\underline{q}$ can be transformed into a recursion in $\underline{\hat{b}}$, i.e.,

$$\underline{\hat{b}}_1 = \underline{P}'\underline{q}_1 = \underline{P}' \left(\phi'\underline{q}_o + \Gamma'\underline{u}\right) = P'\phi'P'^{-1}\,\hat{b}_o + \underline{P}'\underline{\Gamma}'\underline{u}_1 \tag{C.26}$$

$$\triangleq \underline{F}\,\hat{\underline{b}}_o + \underline{G}\,\underline{u}_1 \quad .$$

By substituting the matrix values given above, F and G are given by

$$F = \begin{bmatrix} \frac{2(2N^2+2N-9)}{(2N-1)(2N+3)} & \frac{10\,N(N+1)}{(2N-1)(2N+3)} & -\frac{5\,N(N+1)}{2(2N-1)(2N+3)} \\ -\frac{3}{N(N+1)} & 1 & -\frac{1}{2} \\ \frac{90}{N(N+1)(2N-1)(2N+3)} & -\frac{60}{(2N-1)(2N+3)} & \frac{4(N^2+N+3)}{(2N-1)(2N+3)} \end{bmatrix} \tag{C.27}$$

$$G = \begin{bmatrix} -\frac{3(N-1)}{(2N+1)(2N+3)} & \frac{3(N+2)}{(2N-1)(2N+1)} \\ \frac{3}{(N+1)(2N+1)} & \frac{3}{N(2N+1)} \\ \frac{30}{(N+1)(2N+1)(2N+3)} & -\frac{30}{N(2N-1)(2N+1)} \end{bmatrix} \tag{C.28}$$

Now from Eq. (C.5),

$$\underline{b} = \begin{bmatrix} 1 & & \\ & \Delta & \\ & & \Delta^2 \end{bmatrix} \underline{x} \quad , \tag{C.29}$$

$$\triangleq D\,\underline{x}$$

Thus,

$$\underline{x} = \begin{bmatrix} 1 & & \\ & \Delta^{-1} & \\ & & \Delta^{-2} \end{bmatrix} \underline{b} \quad , \tag{C.30}$$

$$\triangleq D^{-1}\,\underline{b}$$

Therefore,

$$E(\underline{x}\,\underline{x}^T) = E(\underline{D}^{-1}\,\underline{b}\,\underline{b}^T\underline{D}^{-T}) = \underline{D}^{-1}\,\underline{P}'\,\underline{D}^{-T} \qquad (C.31)$$

$$= \begin{bmatrix} \frac{3(3N^3+3N-1)}{(2N-1)(2N+1)(2N+3)} & 0 & \frac{30\,\Delta^{-2}}{(2N-1)(2N+1)(2N+3)} \\ 0 & \frac{3\,\Delta^{-2}}{N(N+1)(2N+1)} & 0 \\ \frac{30\,\Delta^{-2}}{(2N-1)(2N+1)(2N+3)} & 0 & \frac{180\,\Delta^{-4}}{N(N+1)(2N-1)(2N+1)(2N+3)} \end{bmatrix}$$

Going through the transformations, the matrices ϕ and Γ of Eq. (C.1) are calculated to be

$$\underline{\Phi} = \begin{bmatrix} 1 & & \\ & \Delta^{-1} & \\ & & \Delta^{-2} \end{bmatrix} F \begin{bmatrix} 1 & & \\ & \Delta & \\ & & \Delta^2 \end{bmatrix},$$

$$= \begin{bmatrix} \frac{2(2N^2+2N-9)}{(2N-1)(2N+3)} & \frac{10\,N(N+1)\Delta}{(2N-1)(2N+3)} & -\frac{5\,N(N+1)\Delta^2}{2(2N-1)(2N+3)} \\ -\frac{3\Delta^{-1}}{N(N+1)} & 1 & -\frac{\Delta}{2} \\ \frac{90\,\Delta^{-2}}{N(N+1)(2N-1)(2N+3)} & -\frac{60\,\Delta^{-1}}{(2N-1)(2N+3)} & \frac{4\,(N^2+N+3)}{(2N-1)(2N+3)} \end{bmatrix}, \qquad (C.32)$$

and

$$\underline{\Gamma} = \begin{bmatrix} 1 & & \\ & \Delta^{-1} & \\ & & \Delta^{-2} \end{bmatrix} \underline{G} = \begin{bmatrix} -\frac{3(N-1)}{(2N+1)(2N+3)} & \frac{3(N+2)}{(2N-1)(2N+1)} \\ \frac{3\,\Delta^{-1}}{(N+1)(2N+1)} & \frac{3\,\Delta^{-1}}{N(2N+1)} \\ \frac{30\,\Delta^{-2}}{(N+1)(2N+1)(2N+3)} & -\frac{30\,\Delta^{-2}}{N(2N-1)(2N+1)} \end{bmatrix} \qquad (C.33)$$

Equations (C.1), (C.32) and (C.33) thus give a recursive method of determining the smoothed value of $(x_{n+1}, \dot{x}_{n+1}, \ddot{x}_{n+1})$, given the previously smoothed values $(x_n, \dot{x}_n, \ddot{x}_n)$ plus the new measurement y_{n+N+1}.

By combining Eqs. (C.3), (C.5), (C.13), and (C.19), we get the standard deviations of the errors in $(x_n, \dot{x}_n, \ddot{x}_n)$ due to the measurement noise. This is

$$\sigma_{x_n} = \left[\frac{3\,(3N^2 + 3N-1)}{(2N-1)(2N+1)(2N+3)}\right]^{1/2} \sigma \ ,$$

$$\sigma_{\dot{x}_n} = \left[\frac{1}{\Delta^2}\ \frac{3}{N(N+1)(2N+1)}\right]^{1/2} \sigma \ ,$$

$$\sigma_{\ddot{x}_n} = \left[\frac{1}{\Delta^4}\ \frac{180}{N(N+1)(2N-1)(2N+1)(2N+3)}\right]^{1/2} \sigma \ . \qquad \text{(C.34)}$$

If a total of m points is used for smoothing, and we wish to find the midpoint, then we can write

$$m = 2N+1, \text{ or } N = (m-1)/2 \ . \qquad \text{(C.35)}$$

By substituting Eq. (C.35) into (C.34) we obtain

$$\sigma_{x_n} = \left[\frac{3(3m^2-7)}{4m(m^2-4)}\right]^{1/2} \sigma \ ,$$

$$\sigma_{\dot{x}_n} = \left[\frac{12}{\Delta^2\ m(m^2-1)}\right]^{1/2} \sigma \ ,$$

$$\sigma_{\ddot{x}_n} = \left[\frac{4\ (180)}{\Delta^4\ m(m^2-1)(m^2-4)}\right]^{1/2} \sigma \ . \qquad \text{(C.36)}$$

Remark:

One note of caution needs to be stated. The recursive algorithm given by the above development would represent a sizable real time saving since the associated matrices are precomputable; however, it may encounter numerical problems due to computational error build-up. This is caused by the fact that the "transition matrix" ϕ has triple eigenvalues at 1.

APPENDIX D

Finite Memory Non-Linear Interpolation for a Circular-Arc Trajectory

This appendix briefly describes a non-linear interpolation method to identify the unknown turn-rate associated with a circular-arc trajectory. This represents an extension to fixed interval smoothing for a constant acceleration trajectory which was discussed in Appendix C. The basic idea is to apply the fixed-interval, fixed-lag smoothing concept to a case where the underlying system equations contain unknown parameter(s).

Basic Model Equations As mentioned elsewhere, aircraft kinematics under a constant speed and a constant turn rate (e.g., a constant roll angle maneuver with no wind) can be expressed very simply by

$$\frac{d}{dt}\begin{bmatrix}\underline{p}\\ \underline{v}\end{bmatrix} = \begin{bmatrix}0 & \underline{I}\\ 0 & \underline{\Omega}\end{bmatrix}\begin{bmatrix}\underline{p}\\ \underline{v}\end{bmatrix} \tag{D.1}$$

where $\underline{p}$ and $\underline{v}$ are position and velocity vectors $(x,y)^T$ and $(\dot{x},\dot{y})^T$ with respect to an earth-fixed rectangular coordinate system. (Our concern is limited to the horizontal plane; thus, the cross-coupling between the lateral and longitudinal/vertical dynamics is ignored.) The 2 x 2 matrix $\underline{\Omega}$ represents the turn effect and is given by

$$\underline{\Omega} = \begin{bmatrix}0 & \omega\\ -\omega & 0\end{bmatrix} . \tag{D.2}$$

It is noted that when $\omega = 0$, Eq. (D.1) represents constant speed rectilinear motion.

If the radar measurements are taken at a regular interval of time, Δ , Eq. (D.1) can be integrated over a time interval $[n\Delta, (n+1)\Delta]$.

The resulting discrete equation is given by

$$\begin{bmatrix} \underline{p} \\ \underline{v} \end{bmatrix}_{n+1} = \begin{bmatrix} I & \underline{T}_1 \\ 0 & \underline{T} \end{bmatrix} \begin{bmatrix} \underline{p} \\ \underline{v} \end{bmatrix}_n \quad , \tag{D.3}$$

where the $\underline{T}$ and $\underline{T}_1$ matrices are given by

$$\underline{T} = \begin{bmatrix} c\omega\Delta & s\omega\Delta \\ -s\omega\Delta & c\omega\Delta \end{bmatrix}, \quad \underline{T}_1 = \frac{1}{\omega} \begin{bmatrix} s\omega\Delta & 1-c\omega\Delta \\ -1+c\omega\Delta & s\omega\Delta \end{bmatrix}, \tag{D.4}$$

and

$$c\omega\Delta = \cos(\omega\Delta) \; ; \quad s\omega\Delta = \sin(\omega\Delta) \quad .$$

The radar measurements of range (r) and bearing (β) can be expressed as

$$\begin{aligned} r_m &= [x^2 + y^2]^{\frac{1}{2}} + \xi_r \quad , \\ b_m &= \tan^{-1}(y,x) + \xi_b \quad , \end{aligned} \tag{D.5}$$

where ξ_r and ξ_b are measurement errors assuming independent white noise processes. By solving for the position variables, the so-called pseudo-linear observation equations are obtained, i.e.,

$$\begin{aligned} x_m &= r_m \cos b_m \simeq x + \xi_x \quad , \\ y_m &= r_m \sin b_m \simeq y + \xi_y \quad , \end{aligned}$$

or

$$\underline{p}_m = \underline{p} + \underline{\xi}_p \quad . \tag{D.6}$$

It is noted that the "white noise" process $\underline{\xi}_p$ is no longer independent, i.e., $E(\xi_x \xi_y) \neq 0$. In fact the error covariance matrix is given by

$$\underline{R}_n \triangleq E \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} [\xi_x \xi_y] \tag{D.7}$$

$$= \begin{bmatrix} \cos^2 b\, \sigma_r^2 + r^2 \sin^2 b\, \sigma_b^2 & \cos b \sin b\, [\sigma_r^2 - r^2\sigma_b^2] \\ \cos b \sin b\, [\sigma_r^2 - r^2\sigma_b^2] & \sin^2 b\, \sigma_r^2 + r^2 \cos^2 b\, \sigma_b^2 \end{bmatrix} ,$$

where σ_r and σ_b are standard deviations for range and bearing measurement errors.

Auto Regressive Equation Because the pair of equations (D.3) and (D.6) represents an observable system, the first order difference equation in four state variables can be rewritten as a second order difference equation in two state variables. After algebraic manipulation, the following is obtained

$$\underline{p}_{n+1} = (1 + \underline{T})\,\underline{p}_n - \underline{T}\,\underline{p}_{n-1} \quad . \tag{D.8}$$

Equation (D.8) is in the auto-regressive equation form. One major advantage over the standard state space representation of Eq. (D.3) is that Eq. (D.8) involves the state variables which are directly observable through the measurement Eq. (D.6). That is, if $\underline{p}_0$, $\underline{p}_1$ and T are known, then $\underline{p}_2$ can be computed very simply; whereas one needs to solve for $\underline{v}_0$ and $\underline{v}_1$ prior to computing $\underline{p}_2$ using Eq. (D.3).

It is noted that Eq. (D.8) can be rewritten as

$$\underline{p}_{n+1} - \underline{p}_n = T(\underline{p}_n - \underline{p}_{n-1}) \quad , \tag{D.9}$$

which states that the current position difference is a rotation of the previous position difference, confirming our intuition. Furthermore, when the turn-rate, ω is 0, the transformation T reduces to identity; thus, the equation reduces to

$$\underline{p}_{n+1} = 2\,\underline{p}_n - \underline{p}_{n-1} \quad . \tag{D.10}$$

The last equation, of course, represents a straight line, constant speed trajectory.

Further simplification is possible. For example, Eq. (D.8) can be decoupled into two identical scalar third order difference equations of the form

$$x_{n+1} = (1 + 2c\theta)x_n - (1 + 2c\theta)x_{n-1} + x_{n-2} \quad . \tag{D.11}$$

In the following development, Eq. (D.8) is mainly utilized.

Fixed-Interval, Fixed-Lag Interpolation Algorithm Equation (D.8) can be used to formulate a fixed-interval, fixed-lag interpolation problem. The basic problem statement is: Obtain the best estimates of $\underline{p}_0$ and $\underline{p}_1$, given a set of measurements $(p_{km}: \quad k = -N+1,\ldots,N)$. One of the shortest ways to derive an interpolation algorithm is to write out the relationship between the measurements and the variables to be estimated. Thus, we seek a set of equations of the form

$$\underline{p}_{km} = \underline{A}_k \underline{p}_1 + \underline{B}_k \underline{p}_0 + \underline{\xi}_{p,k}, \qquad k = -N+1,\ldots,0,1,\ldots,N. \tag{D.12}$$

Equation (D.12) can be written more compactly by using matrix notation as follows.

$$\underline{P}_N = \underline{A}(\omega)\, \underline{\theta} + \underline{n}, \tag{D.13}$$

where

$$\underline{P}_N = \begin{bmatrix} p_{Nm} \\ p_{N-1m} \\ \vdots \\ p_{1m} \\ p_{0m} \\ \vdots \\ p_{-N+1m} \end{bmatrix}, \quad \underline{\theta} = \begin{bmatrix} \underline{p}_1 \\ \underline{p}_0 \end{bmatrix}, \quad \underline{n} = \begin{bmatrix} \underline{\xi}_{p,N} \\ \underline{\xi}_{p,N-1} \\ \vdots \\ \underline{\xi}_{p,1} \\ \underline{\xi}_{p,0} \\ \vdots \\ \underline{\xi}_{p,-N+1} \end{bmatrix}$$

and

$$\underline{A}(\omega) = \begin{bmatrix} \underline{A}_N & \underline{B}_N \\ \underline{A}_{N-1} & \underline{B}_{N-1} \\ \vdots & \vdots \\ \underline{A}_1 & \underline{B}_1 \\ \underline{A}_0 & \underline{B}_0 \\ \vdots & \vdots \\ \underline{A}_{-N+1}, & \underline{B}_{-N+1} \end{bmatrix}. \tag{D.14}$$

We can take advantage of a certain symmetry of the difference equation. The backward difference equation is given by

$$\underline{p}_{n-1} = \underline{T}^{-1}(1+\underline{T})\underline{p}_n - \underline{T}^{-1}\underline{p}_{n+1} ,$$

or using the orthogonality property ($\underline{T}^{-1} = \underline{T}^t$)

$$\underline{p}_{n-1} = (1+\underline{S})\underline{p}_n - \underline{S}\, p_{n+1} . \tag{D.15}$$

Thus, the forward and backward equations look alike except the appearance of the transpose. Now $\underline{A}(\omega)$ can be written explicitly as

$$A(\omega) = \left[\begin{array}{c|c} 1+T+\ldots+T^{N-1} & -(T+T^2+\ldots+T^{N-1}) \\ 1+\ldots+T^{N-2} & -(\ T\ +\ldots+T^{N-2}) \\ \ldots & \ldots \\ \ldots & \ldots \\ 1+T & -T \\ I & 0 \\ \hline 0 & I \\ -S & 1+S \\ \ldots & \ldots \\ \ldots & \ldots \\ -(S+\ldots+S^{N-2}) & 1+S+\ldots+S^{N-2} \\ -(S+\ldots+S^{N-1}) & 1+S+\ldots+S^{N-1} \end{array}\right] , \tag{D.16}$$

where

$$\underline{S} \triangleq \underline{T}^{-1} = \underline{T}^t .$$

If the turn rate, ω, is known, then the optimal estimate of $\underline{\theta}$ is given by the usual weighted least squares solution:

$$\underline{\hat{\theta}}(\omega) = [\underline{A}^T(\omega)\ \underline{\Sigma}_\eta^{-1}\ \underline{A}(\omega)]^{-1}\ [\underline{A}^T(\omega)\ \underline{\Sigma}_\eta^{-1}\ \underline{P}_N] . \tag{D.17}$$

Here, $\underline{\Sigma}_\eta$ is the aggregate covariance matrix (assumed known) of the measurement errors, i.e.,

$$\underline{\Sigma}_{\eta} = \text{Diag}\,[\underline{R}_N, \underline{R}_{N-1}, \ldots, \underline{R}_1, \underline{R}_0, \ldots, \underline{R}_{N+1}] \,.$$

One measure of the estimation accuracy is given by the weighted rms error

$$J(\omega) = \frac{1}{2} \left|\left| \underline{P}_N^m - \underline{A}(\omega)\, \underline{\hat{\theta}}(\omega) \right|\right|^2_{\underline{\Sigma}_{\eta}^{-1}} \,, \tag{D.18}$$

where

$$\left|\left| \underline{V} \right|\right|_{\underline{R}^{-1}} = [\underline{V}^T \underline{R}^{-1} \underline{V}]^{\frac{1}{2}} \,.$$

Equations (D.17) and (D.18) form the basis for a non-linear interpolation algorithm when the turn-rate is not known. The algorithm is motivated by the so-called bank of Kalman filter approach to the parameter identification/state estimation problem. The idea is depicted in Fig. D-1 which can be easily implemented in parallel micro-processor architecture.

The past measurements and error covariance matrices are stored in stacks of "push-down" memories which are directly accessible to each computational module. The matrices $\underline{A}(\omega_i)$'s are stored in read-only memories local to the micro-processor. Given these data sets, each processor works on its local computation independently of others to generate the estimate and the associated rms error. When the individual processor finishes, the results are sent to a comparator module which chooses a processor with the smallest rms error.

There are several advantages of the above scheme. These are:

(1) The computational structure is simple and modularized so that additional modules can be accommodated easily;

(2) Computational speed is independent of discretization of the parameter space, i.e., number of ω_i's; and

(3) Each processor would have an identical program. The only distinction lies in the values of variables in the local read-only memories.

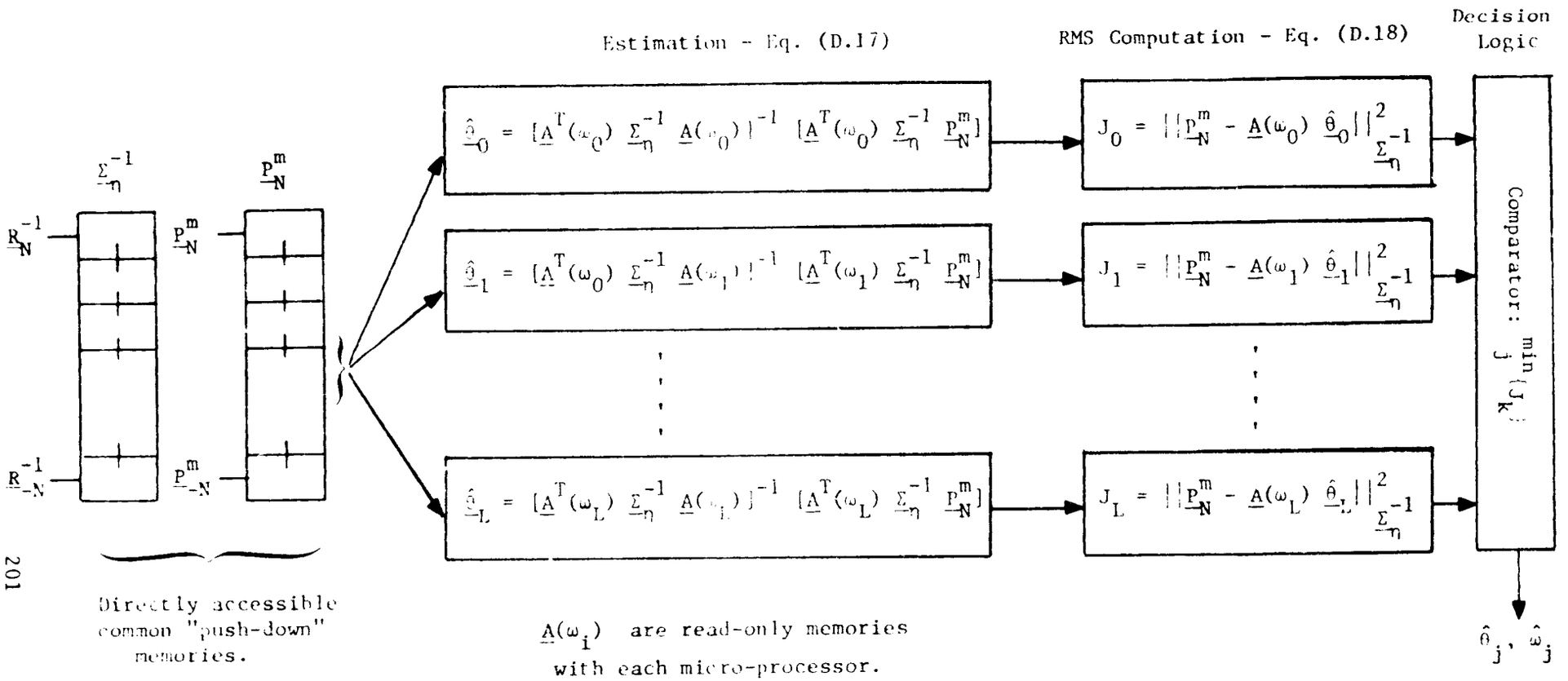


Figure D-1. Computational Block Diagram for a Nonlinear Interpolation Algorithm Based on a Parallel Micro-Processor Architecture.

One major disadvantage is that each module cannot share overhead functions (e.g., a 4 x 4 matrix inversion routine) because it must be essentially self-contained. However, memory requirements are becoming less and less severe today due to technical advances in the computer industry.

Critique In the previous sections and in Appendix C, interpolation algorithms are developed based on certain kinematic models. Essentially smoothed state estimates are obtained which are "lagging" by a fixed interval of time compared to the current reference time. Moreover, the proposed algorithms are based on the batch processing of data spanning a finite time interval. Thus, the smoothed estimates will exhibit similar characteristics which are also encountered by filtering problems. For example, if the time interval is longer, then the high frequency error effect on the estimate diminishes, but the error due to an inaccurate model increases. This implies that the time interval, high frequency error magnitude and modeling accuracy are intimately related to the achievable smoothing accuracy. Therefore, the smoothing interval and dynamic model must be chosen carefully. It is well known that the same statement applies to the filtering problem if the "time interval" is replaced by the "inverse of filter bandwidth (or gain)".

REFERENCES

1. Kelly, J. R., and D. Williams, "Factors Affecting In-trail Following Using CDTI," 18th Annual Manual Control Conference, Dayton, OH, June 1982.

2. Goka, T., "Enhanced TCAS II/CDTI Traffic Sensor Digital Simulation Model and Program Description," NASA CR-172445, October 1984.

3. Goka, T., and J. A. Sorensen, "New Aircraft Tracking Algorithms Based on DABS Sensors," AIAA Paper No. 81-1800, Presented at 1981 AIAA Guidance and Control Conference, Albuquerque, NM, August 1981.

4. Thomas, H. W., et al, "Use of Aircraft-Derived Data to Assist in ATC Tracking Systems - Part 1: Accuracy and Theoretical Considerations," IEE Proc., Vol. 129, Pt F, No. 4, August 1982.

5. Sinsky, A. I., "Alpha-Beta Tracking Errors for Orbiting Targets," Bendix Report No. BCD-TN-81-033 (FB-467-012), The Bendix Corp., Baltimore, MD, June 1981.

6. Koenke, E. J., et al, "FAA BCAS Concept - Concept Description," FAA Report No. FAA-EM-78-5, April 1978.

7. Cohen, S. and F. Maginnis, "Statistical Summary for Los Angeles Basin Standard Traffic Model." The Mitre Corp., Report No. MTR-6387, McLean, VA, April 1973.

8. Castella, F. R., "An Adaptive Two-Dimensional Kalman Tracking Filter," IEEE Trans. Aerosp. Electron. Syst., Vol. AES-16, No. 6, November 1980.

9. Anon., "Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System (TCAS) - Final Draft" RTCA Paper No. 106-83/SC 147-120, Radio Technical Commission for Aeronautics, Washington, DC, March 1983.

10. Billmann, B., et al, "Modeling Active Beacon Collision Avoidance System (BCAS) Measurement Error: An Empirical Approach," FAA Report No. FAA-RD-80-83, FAA Technical Center, Atlantic City, NJ, May 1980.

11. Tarn, T. J., and J. Zaborsky, "A Linear Filter for Discrete Systems with Correlated Measurement Noise," in Pattern Recognition and Machine Learning, K. S. Fu (ed), Plenum Press, 1971.

References (Continued)

12. Goka, T., "An Improved Active BCAS Range/Range Rate Tracker Algorithm," AMA Report No. 82-17, Analytical Mechanics Associates, Inc., Mt. View, CA, April 1982.

13. Andrews, J. W., "An Improved Technique for Altitude Tracking of Aircraft," FAA Report No. FAA-RD-82-14, Lincoln Laboratory, Lexington, MA, March 1981.

14. Billmann, B. R., "Analysis of a Nonlinear Altitude Tracking Method," FAA Report No. FAA-RD-81-60A, FAA Technical Center, Atlantic City, NJ, February 1982.

15. Benedict, T. R. and Bordner, G. W., "Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations," IRE Transactions on Automatic Control, July 1982.

16. Miller, R. W., "Assympototic Behavior of the Kalman Filter with Exponential Aging," AIAA Journal, Vol. 9, No. 3, pp 537-539, 1971.

17. Tarn, T. J. and Zaborsky, J., "A Practical Nondiverging Filter," AIAA Journal, Vol. 8, No. 6, June 1970.

18. Box, G. E., and G. M. Jenkins, Time Series Analysis - Forecasting and Control, Holden-Day, Inc., San Francisco, CA, June 1969.

19. Anon., "Full BCAS Critical Design Review for the Engineering Model - Task I," Bendix Corp., Communications Division, Baltimore, MD, September 1981.

20. Williams, D. H., and G. C. Moen, "Simulation Study of Traffic-Sensor Noise Effects on Utilization of Traffic Situation Display for Self-Spacing Task," NASA TP-2082, February 1983.

21. Sinsky, A. I., and B. J. Tier, "Coordinate Trnasformation Algorithms," Report No. BCD-TN-82-020, Bendix Corp., Baltimore, MD, May 1982.

22. Sobocinski, R., "Flight Test Results with Collision Avoidance Systems," Paper No. 20-2. Proceed. of 5th Digital Avionics Systems Conf., Seattle, WA, October 1983.

23. Reed, J. E., "Enhanced TCAS II Test Results," Paper No. 20-3, Proceed. of 5th Digital Avionics Systems Conf., Seattle, WA, October 1983.

24. Kelly, J. R., "Effect of Lead-Aircraft Ground-Speed Quantization on Self-Spacing Performance Using a Cockpit Display of Traffic Information," NASA TP-2194, October 1983.

25. Sorensen, J. A., and T. Goka, "Analysis of In-trail Following Dynamics of CDTI-Equipped Aircraft," AIAA Paper No. 82-1330, Presented at 1982 AIAA Atmospheric Flight Mechanics Conference, San Diego, CA, August 1982.

References (Continued)

26. Sorensen, J. and T. Goka, "Prediction of Cockpit Simulator Experimental Outcome Using System Models," Presented at 20th Annual Manual Control Conference, Sunnyvale, CA, June 1984.

27. Gelb, A. (ed), Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.

28. Graupe, D., "On Convergence of Least-Squares Identifiers of Auto-Regressive Models Having Stable and Unstable Roots," IEEE Trans. Auto. Cont., Vol. AC-25, No. 5, October 1980.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR - 172444		

4. Title and Subtitle	5. Report Date
Analysis of Estimation Algorithms for CDTI and CAS Applications	April 1985
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Tsuyoshi Goka	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Analytical Mechanics Associates, Inc. 2483 Old Middlefield Way, Suite 210 Mountain View, California 94043	NAS1-16135
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code
	505-35-13-03

15. Supplementary Notes

Langley Technical Monitors: Roland L. Bowles and James R. Kelly
Final Report

16. Abstract

The objectives of this project were to analyze and/or to develop estimation algorithms for Cockpit Display of Traffic Information (CDTI) and Collision Avoidance System (CAS) applications. The algorithms are based on actual or projected operational and performance characteristics of an Enhanced TCAS II traffic sensor developed by Bendix and the Federal Aviation Administration.

Three algorithm areas are examined and discussed. These are horizontal x and y, range and altitude estimation algorithms. Raw estimation errors are quantified using Monte Carlo simulations developed for each application; the raw errors are then used to infer impacts on the CDTI and CAS applications. Applications of smoothing algorithms to CDTI problems are also discussed briefly.

Technical conclusions are summarized based on the analysis of simulation results.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
CDTI Enhanced TCAS II Horizontal Estimator Mode C Altitude Estimator	Unclassified - Unlimited Subject Category 06

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	219	A10

* 30*

For sale by the National Technical Information Service Springfield Virginia 22161